FAULT-TOLERANCE ON NEAR-TERM QUANTUM COMPUTERS AND SUBSYSTEM QUANTUM ERROR CORRECTING CODES

A Dissertation
Presented to
The Academic Faculty

By

Muyuan Li

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computational Science and Engineering

Georgia Institute of Technology

May 2020

FAULT-TOLERANCE ON NEAR-TERM QUANTUM COMPUTERS AND SUBSYSTEM QUANTUM ERROR CORRECTING CODES

Approved by:

Dr. Kenneth R. Brown, Advisor
Department of Electrical and Computer Engineering
Duke University

Dr. Edmond Chow
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. T.A. Brian Kennedy
School of Physics
Georgia Institute of Technology

Dr. C. David Sherrill
School of Chemistry and Biochemistry
Georgia Institute of Technology

Dr. Richard Vuduc
School of Computational Science and Engineering
Georgia Institute of Technology

Date Approved: March 19, 2020

I think it is safe to say that no one understands quantum mechanics.

R. P. Feynman

To my family and my friends.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Ken Brown, who has guided me through my graduate studies with his patience, wisdom, and generosity. He has always been supportive and helpful, and always makes himself available when I needed. I have been constantly inspired by his depth of knowledge in research, as well as his immense passion for life.

I would also like to thank my committee members, Professors Edmond Chow, T.A. Brian Kennedy, C. David Sherrill, and Richard Vuduc, for their time and helpful suggestions.

One half of my graduate career was spent at Georgia Tech and the other half at Duke University. I am fortunate enough to have met good friends and colleagues at both places.

I want to thank my fellow CSE cohort: Frederick Hohman, Aradhya Biswas, Amrita Gupta, David Robinson, and Jordi Wolfson-Pou. They have given me a great amount of help with my classes and exams for the time that I was at Georgia Tech as well as random questions after I have moved to Duke. I offer my thanks to Jordi for inviting me into his fantasy basketball league.

I would like to thank the entire Brown Lab. It was a fantastic group to spend a graduate career with. I wish to thank Mauricio Gutiérrez and Colin Trout, for teaching me the fundamentals of quantum error correction when I first started. I especially want to thank Dripto Debroy, Michael Newman, Natalie Brown, and Shilin Huang for all the great conversations about quantum error correction and wonderful things in life. I am indebted to Michael Newman for putting up with my ignorance and teaching me important things in quantum information.

I would also like to thank Dr. Jungsang Kim and the MIST group at Duke University. I would never have learnt so much about ion trap quantum computing without them.

I would like to thank the IBM graduate internship program for having me visit the IBM T.J. Watson research center. I especially want to thank Andrew Cross and Ted Yoder for

their help during my time there.

Graduate school would have been so much more difficult without good company and delicious Chinese food. I thank Bichen Zhang, Zhubing Jia, Shilin Huang, and Lu Qi for helping me satisfy my Chinese stomach on weekends and during holidays.

Finally, I want to thank my parents Jing Li and Xuming Huang, for allowing me to pursue my own dreams and believing in me every step of the way. I could never have achieved anything without them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Large-scale fault-tolerant quantum computers are supposed to provide exponential speedup over many classical algorithms for solving realistic computationally intensive problems. Given that practical quantum computers are extremely sensitive to noise, error correction protocols have to be employed in order to suppress noises in the quantum system and maintain the fidelity of computations. As quantum computing experiments are progressing into the regime where active quantum error correction can soon be implemented, it is becoming more important to understand the properties of small quantum error correcting codes and how to efficiently implement them in actual experiments given realistic near-term quantum device specifications.

In this dissertation, we present our work on suppressing errors in quantum computing systems in two directions: realizing fault-tolerance with several small quantum error-correcting codes under realistic device assumptions, and developing large-scale code families with subsystem quantum error correcting codes. For small codes such as the Bare [[7,1,3]] code, the Bacon-Shor code, and the rotated 17-qubit surface code, we show designs for implementing them in realistic ion trap quantum computers while minimizing the amount of resources needed and optimizing fidelity of the logical system. For families of subsystem quantum error correcting codes, we investigate the compass codes, Bravyi-Bacon-Shor codes and subsystem hypergraph product codes, and show that these codes can be advantageous in terms of handling biased noises, having good code parameters, and exhibiting efficient decoding performance.

CHAPTER 1
INTRODUCTION

Despite of the massive advertisement by media through all possible channels, large-scale fault-tolerant quantum computer that can execute advanced algorithms still seems like a distant dream to most people. While physicists and engineers are making tremendous progress towards building practical quantum computers on different platforms, the path leading to a reliable large-scale quantum computer remains impeded by the abundance of noise in the quantum system due to decoherence of quantum states and imperfect controls. Unlike classical bits which are represented by multiple particles and are robust to noises due to the statistics of their collective behavior, quantum bits (qubits) often experience single-particle operations that are prone to errors. Furthermore, interactions between qubits and their environment can lead to decay of information over time. Therefore it is crucial to include a layer of fault-tolerant protocols when designing quantum computers in order to suppress the effect of environmental and operational noise on the quantum computations being carried out.

In this thesis we focus on using active quantum error correction to achieve fault-tolerance in realistic quantum computers. Since errors present in quantum computers are caused by certain physical sources, a comprehensive understanding of these physical noise sources can provide guide to adapt theoretical error correcting codes to the realistic environment that the quantum computer is built in. Given that we are already capable of building noisy intermediate-scale quantum devices with tens of qubits, it is certainly beneficial to utilize our understanding of the physical system of these near-term quantum computers to design fault-tolerant protocols that are well-performing and resource efficient. Furthermore, when large-scale quantum computers are considered, it is essential to build error correction protocols into the architecture of the device, so that there is a uniform method of realiz-

ing fault-tolerant universal quantum operations throughout the system. For that purpose, a good family of quantum error correcting code should be chosen given restrictions and requirements of the device. In this thesis we will investigate how does realistic noise and constraints present in near-term trapped ion quantum computers affect fault-tolerance of small quantum error correcting codes, as well as explore new families of subsystem quantum error correction codes.

1.1 Organization of This Thesis

This thesis presents studies on quantum error correction from two perspectives: small quantum error correcting codes that are implementable on near-term quantum computers (Chapters 3, 4, and 5), and large families of subsystem quantum error correcting codes (Chapters 6).

This dissertation is divided into seven chapters. In Chapter 1 we briefly review the basics of quantum information and quantum error correction. In Chapter 2 we discuss a sampling algorithm that we have implemented and used to efficiently study the numerical performance of quantum error correcting codes. In Chapter 3 we study the Bare $[\![7, 1, 3]\!]$ code and show that requirements of fault-tolerance change according to the error model being considered. In Chapter 4, we present methods to fault-tolerantly and efficiently perform syndrome extraction for the distance-3 Bacon-Shor code when the underlying 2-dimensional quantum compass model is considered, and show that the distance-3 Bacon-Shor code can outperform the distance-3 rotated surface code when a realistic metric is considered. In Chapter 5 we numerically benchmark performances of several distance-3 quantum error correcting codes in a realistic ion trap environment, and present an algorithm that can suppress ion trap crosstalk errors to the second order when some error correction codes are considered. In Chapter 6 we study families of subsystem quantum error correcting codes that can be constructed using classical expander codes: the Bravyi-Bacon-Shor codes and subsystem hypergraph product codes. Finally in Chapter 7 we summarize the

results, and give an overview of potential directions for future work.

1.2 Quantum Information and Quantum Computation

1.2.1 Quantum States

In contrast to classical computers which stores and processes information in the form of binary data, quantum computers work through manipulating and processing quantum bits (qubits) that are superpositions or probability ensembles of basis quantum states [1]. The information or state of a single qubit can be described using a vector in $\mathbb{C}^2$,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1.1}$$

where $|\alpha|^2 + |\beta|^2 = 1$. The basis states $|0\rangle = [1\ 0]^T$ and $|1\rangle = [0\ 1]^T$ represent the two classical values that could be obtained when the qubit is measured, and the probability of obtaining the two values is given by the coefficients $|\alpha|^2$ and $|\beta|^2$. A single qubit can be in the superposition of two states $|0\rangle\,,\ |1\rangle$, two qubits can be in the superposition of four states $|00\rangle\,,\ |01\rangle\,,\ |10\rangle\,,\ |11\rangle$. Consequently n qubits can be in the superposition of 2^n basis states, where n-qubit pure states $|\psi\rangle$ can be represented by taking the tensor product of individual states. This is the core reason for quantum computing to potentially offer exponential speedup over classical computing algorithms.

A single qubit quantum state can be expressed as a density operator, which is a positive semi-definite matrix with trace of one. A single qubit quantum state density operator ρ acts on the state space of the qubit, $\mathbb{C}^2$, can be denoted as $\rho(\mathbb{C})$. Similarly, an n-qubit quantum state can be define as a density operator acting on the tensor product of the state spaces of n individual qubits, $\rho((\mathbb{C}^2)^{\otimes n})$.

A general qubit state that is not necessarily pure ($|\alpha|^2 + |\beta|^2 \leq 1$) can be represented

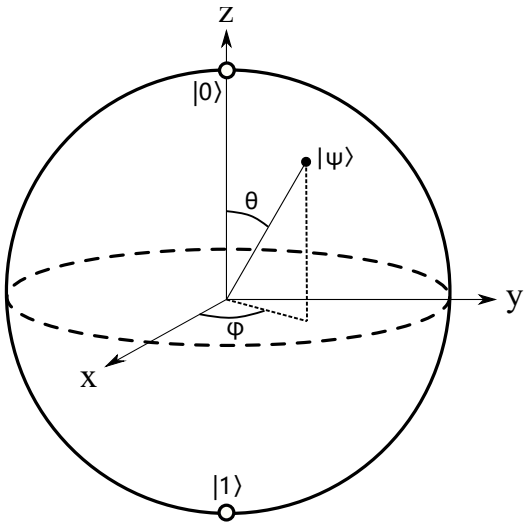


Figure 1.1: A pure single-qubit state can be represented as a point on the Bloch sphere with $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$.

by a sum of Pauli matrices and identity

$$\rho = \frac{1}{2}(I + \vec{r}\cdot\vec{\sigma}) = \frac{1}{2}(I + r_x X + r_y Y + r_z Z) \tag{1.2}$$

where $\vec{r}$ is a real-valued vector that is used to define the qubit state. For pure states $|\vec{r}| = 1$ and for mixed states $|\vec{r}| \leq 1$. When the Bloch sphere representation as shown in Figure 1.1 is considered, pure states are represented as points on the surface of the Bloch sphere while mixed states as represented as points within the surface of the Bloch sphere.

A quantum state ρ is mixed if and only if $\text{Tr}(\rho^2) \neq 1$. Since density operator ρ can be diagonalized as

$$\rho = \sum_i a_i |\psi_i\rangle \langle\psi_i| \tag{1.3}$$

where $\langle\psi_i|\psi_j\rangle = 0$ for $i \neq j$, and a_i are nonnegative and sum to one. Therefore, any density operator can be thought of as a probabilistic mixture of orthogonal pure states.

A quantum state ρ is separable if it can be decomposed as

$$\rho = \sum_i a_i (A_i \otimes B_i), \tag{1.4}$$

where the a_i are nonnegative and sum to one where A_i and B_i are density matrices on the individual partitions. If not, this quantum state ρ is entangled. If ρ is a pure state, then it is separable if and only if it can be identified as a direct tensor product in its state space, or otherwise it is entangled.

Quantum states can be manipulated and accessed through quantum operations. For a pure quantum state in space $\mathcal{H}$, evolution is performed using the unitary U,

$$|\psi\rangle \to U\,|\psi\rangle \tag{1.5}$$

where $UU^\dagger = I$. Similar to quantum states that can be combined as tensor products of different states, quantum operations can also be combined using tensor products as the Kronecker product of matrices. In a more general sense, quantum channels can be used to describe mappings between general quantum states. A quantum channel is defined as a completely-positive trace-preserving map. These properties ensure that a quantum channel maps density operators to density operators. Furthermore, Choi's theorem [2] on completely positive trace-preserving maps states that any $\Phi : \mathbb{C}^{n\times n} \to \mathbb{C}^{n'\times n'}$ can be written as

$$\Psi(A) = \sum_{i=1}^{N} K_i A K_i^\dagger, \tag{1.6}$$

where $N \leq nn'$ and $\sum_i K_i K_i^\dagger = I$. The set of operators $\{K_i\}$ are called the Kraus operators of the underlying quantum channel.

A positive-operator valued measure (POVM) is a collection of positive semidefinite operators $\{M_i\}$ acting on a Hilbert space satisfying $\sum_i M_i = I$. In the case where $\{M_i\}$ are orthogonal projectors, we call the set of operators $\{M_i\}$ a projective measurement, which acts on a quantum state ρ as returning a classical output i with probability $\text{Tr}(M_i\rho)$. Conditioning on outcome i, the projective measurement maps ρ to $M_i\rho/\text{Tr}(M_i\rho)$.

1.2.2 Quantum Operations and Quantum Gates

Through operations that precisely control quantum phenomena and manipulate quantum states, it is in principle possible for quantum algorithms to outperform their classical counterparts. These quantum operations are generally referred to as quantum gates. A quantum gate that operates on n-qubits can be described using a $2^n \times 2^n$ unitary matrix. The single qubit Pauli group $\mathcal{P} = \{I, X, Y, Z\}$ is the group of unitary matrices generated by the 2×2 identity matrix I and the Pauli X, Y, and Z matrices:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.7}$$

In the computational basis with basis states $|0\rangle$ and $|1\rangle$, the Pauli X gate is similar to a classical NOT gate. The Pauli Z gate flips the phase of the qubit.

The Pauli group can be generalized to the n-qubit Pauli group $\mathcal{P}^n = \{cP_1 \otimes \ldots \otimes P_n : P_i \in \mathcal{P}, c \in \{1, -1, i, -i\}\}$. Consequently, the n-qubit Clifford group $\mathcal{C}^n$ can be defined as the normalizer of $\mathcal{P}^n$ in $U(2^n)$. The Clifford group is generated by the unitaries

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1.8}$$

For the two-qubit controlled gate CNOT, operation can be applied on the bottom (target) qubit conditioning on the state of the top (control) qubit.

When the circuit model of quantum computation is considered, operations being performed on the target quantum system over time can be represented by quantum circuit. Quantum circuits, serving as the quantum analogue to classical circuit diagrams, are pictorial representations of quantum gates and measurements being performed on a set of qubits.

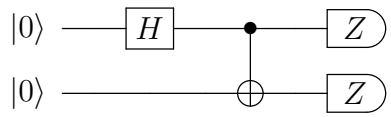


Figure 1.2: Quantum circuit for preparing the state $|\psi\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ with a *Hadamard* and a *CNOT* gate and measuring in the Pauli Z basis.

The Gottesman-Knill theorem [3] states that any Clifford circuit of size n and m gates is classically simulable in polynomial time on a probabilistic classical computer. Furthermore, a get set containing the Clifford group along with any non-Clifford gate is sufficient to achieve universal quantum computation, denoted as a universal gate set. That is, an arbitrary unitary operation U can be approximated up to any tolerance δ using a bounded number of elementary quantum gates in a universal gate set. There are some popular choices of quantum gates that can supplement the Clifford group and achieve universality. These are the CCZ gate,

$$CCZ : |a, b, c\rangle \rightarrow (-1)^{abc} |a, b, c\rangle , \tag{1.9}$$

the *Toffolli* gate,

$$\textit{Toffoli} : |a, b, c\rangle \rightarrow |a, b, (c \oplus ab)\rangle , \tag{1.10}$$

and the T gate

$$T : |a\rangle \rightarrow e^{ai\pi/4} |a\rangle . \tag{1.11}$$

It has also been shown by [4] that $\{\textit{Toffoli}, H\}$ constitutes a universal gate set for quantum computing.

1.3 Quantum Error Correction

While a wide variety of physical systems, such as trapped ions [5, 6], photons [7, 8], superconducting circuits [9, 10], defects in diamonds [11, 12], quantum dots [13, 14] and neutral atoms [15, 16], are being extensively explored as potential candidates for large scale

quantum computation, none of these technologies is universally recognized as the preferred quantum device platform. A problem faced by all of these proposed qubit platforms is the effect of noise on the quantum information being processed and stored.

Information in classical computers are typically realized through the on and off states of transistor switches, so small number of electrons traveling in the wrong direction does not affect the overall performance. This provides robustness for classical bits at the physical level. However in quantum computers where qubits are realized using fragile quantum systems we do not have such robustness against errors. Thus, quantum computers generally require some active error correction procedure in order to ensure the correctness of computation.

1.3.1 Quantum Noise Processes

When implementing a target quantum channel $\mathcal{C}$, due to imperfections of control operations and unwanted interactions with the environment the actual quantum channel being implemented is $\mathcal{E} \circ \mathcal{C}$, where $\mathcal{E}$ represents the noise processes acting on the quantum system that could corrupt the quantum information. When the noise processes are assumed to be independent and identically distributed (i.i.d) on individual physical qubits and also incoherent or stochastic, they can be modeled as a probabilistic ensemble of local Pauli operations. Under these assumptions, two simple noise models that are fundamental to the study of quantum error correction can be written as

1. **Bit-flip Channel**

$$\mathcal{E}_b(\rho) = (1-p)I\rho I + pX\rho X \tag{1.12}$$

2. **Depolarizing Channel**

$$\mathcal{E}_{dep}(\rho) = (1-p)I\rho I + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z) \tag{1.13}$$

where both channels are described by the parameter p, the physical error rate of the chan-

nel. Note that the Bit-flip channel can be easily switched with the Phase-flip channel by replacing Pauli X with Pauli Z operators.

These probabilistic error models have been widely used in quantum error correction literature due to the ability of can be efficiently simulated classically, and that they can characterize the physical error processes accurately when certain assumptions about the quantum system are met, such as when the dynamics of the environment evolves faster than the dynamics of the computation.

However, not all error channels can be efficiently described by the probabilistic approach as defined above. These errors are usually directly motivated by physical processes in the realistic environment, such as errors in the form of a unitary evolution of the computational quantum state. For example, a single-qubit Pauli X overrotation error due to imperfect laser control in an ion trap quantum computer can be characterized as the coherent rotation of angle ϵ round the Pauli X axis in the single-qubit Bloch sphere, written as $R_X(\epsilon) = e^{-i\epsilon X/2}$. Such errors are often referred to as coherent errors, and are of particular interest due to their direct translation from physical systems and severe impact on fidelity of quantum computation. A direct modeling of coherent errors would require the use of density matrix simulators, which scales exponentially in the number of qubits being simulated. In our studies presented in this thesis, we approximate these coherent errors in terms of Pauli matrices, so that the effect of these errors on error correction systems of interest can be efficiently simulated via the Gottesman-Knill theorem. For the example of $R_X(\epsilon)$ mentioned above, we approximated it with the channel $\mathcal{E}_{R_X}(\rho) = (1-p)I\rho I + pX\rho X$, where $p = sin^2(\epsilon)$.

1.3.2 Quantum Error Correcting Codes

When using quantum error correcting codes to protect information embedded in physical qubits, we encode the information into some degrees of freedom of a larger system of physical qubits. We can define a $[\![n, k, d]\!]$ quantum code that uses n physical qubits to

encode k qubits of quantum information and can correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors. This is saying that our quantum information is encoded into a subspace $\mathbb{C}$ of dimension 2^k within a larger Hilbert space $\mathcal{H}$ of dimension 2^n to be protected against errors. The parameter d, denoted the distance of the quantum code, is defined through the Knill-Laflamme error-correction criterion [17].

In order for an error correcting code to independently correct two errors E_a and E_b, it must always be able to distinguish the effect of error E_a acting on a basis codeword $|\psi_a\rangle$ from the effect of error E_b acting on a different basis codeword ψ_b. This is being ensured by enforcing $E_a\psi_a$ to be orthogonal to $E_b\psi_b$

$$\langle\psi_a|E_a^\dagger E_b|\psi_b\rangle = 0 \tag{1.14}$$

where $i \neq j$ and errors E_a and E_b are both correctable errors to the underlying code. Furthermore, in order to ensure that the superposition of the basis states are not disrupted by measurements performed to learn about the errors, we have to guarantee that

$$\langle\psi_i|E_a^\dagger E_b|\psi_i\rangle = \langle\psi_j|E_a^\dagger E_b|\psi_j\rangle \tag{1.15}$$

for all i and j. Therefore, a code can correct all errors $E_a, E_b \in \mathcal{E}$ if and only if it satisfies

$$\langle\psi_i|E_a^\dagger E_b|\psi_j\rangle = C_{ab}\delta_{ij}, \tag{1.16}$$

for all basis codewords $|\psi_i\rangle$ and $|\psi_j\rangle$, and C_{ab} is independent of i and j.

The most popular and easy to describe genre of quantum codes falls into the category of stabilizer codes [18]. A stabilizer group S is an Abelian subgroup of the n-qubit Pauli group $\mathcal{P}^n$ excluding $-I$. The code space $\mathbb{C}$ defined by the stabilizer group $\mathcal{S}$ can be written as

$$C = \{|\psi\rangle : g\,|\psi\rangle = |\psi\rangle \;\forall\, g \in \mathcal{S}\}. \tag{1.17}$$

For a code that uses n qubits to encode k logical qubits of information, the code space $\mathbb{C}$ has 2^k dimension and $\mathcal{S}$ has 2^{n-k} non-trivial generators. For a state $|\psi_i\rangle \in \mathbb{C}$, if $s \in \mathcal{S}$ and $\{s, E\} = 0$, then $sE\,|\psi_i\rangle = -E\,|\psi_i\rangle$ and

$$\langle\psi_i|E|\psi_j\rangle = \langle\psi_i|sE|\psi_j\rangle = -\langle\psi_i|E|\psi_j\rangle = 0. \tag{1.18}$$

Therefore, if $E_a^\dagger E_b$ anticommutes with some element of $\mathcal{S}$ for all errors $E_a, E_b \in \mathcal{E}$, then the stabilizer code defined by the stabilizer group $\mathcal{S}$ will correct the set of errors $\mathcal{E}$.

The set of elements in $\mathcal{P}^n$ that commute with all of $\mathcal{S}$ is defined as the centralizer $C(\mathcal{S})$ of $\mathcal{S}$ in $\mathcal{P}^n$. Note that given the properties of $\mathcal{S}$ and $\mathcal{P}^n$, the centralizer $C(\mathcal{S})$ is equal to the normalizer $\mathcal{N}(\mathcal{S})$. For every $E_L \in \mathcal{N}(\mathcal{S})\backslash\mathcal{S}$, we can see that E_L maps codewords to codewords within the code space $\mathbb{C}$: if $s \in \mathcal{S}$ and $|\psi\rangle \in \mathbb{C}$, then

$$sE_L\,|\psi\rangle = E_L s\,|\psi\rangle = E_L\,|\psi\rangle\,, \tag{1.19}$$

so $E_L\,|\psi\rangle \in \mathbb{C}$. The effect of E_L is nontrivial but will be undectable by the stabilizer code. The operators $E_L \in \mathcal{N}(\mathcal{S})\backslash\mathcal{S}$ are denoted as the logical operators on the code space.

In summary, a stabilizer code $\mathcal{S}$ will detect all errors E that are either in $\mathcal{S}$ or anticommute with some element of $\mathcal{S}$, $E \in \mathcal{S} \cup (\mathcal{P}^n\backslash\mathcal{N}(\mathcal{S}))$. The code will correct any set of errors $\{E_i\}$ if and only if $E_a E_b \in \mathcal{S} \cup (\mathcal{P}^n\backslash\mathcal{N}(\mathcal{S}))\ \forall\ E_a, E_b$. The code distance is d if and only if $|E| \geq d\ \forall\ E \in \mathcal{N}(\mathcal{S})\backslash\mathcal{S}$. If $\mathcal{S}$ contains elements of weight less than d, then it is a degenerate code; otherwise it is a nondegenerate code.

The error syndromes of a stabilizer code is defined as a $(n-k)$-bit string $f(E) = (f_{s_1}(E), \ldots, f_{s_{n-k}}(E))$, where $s_1, \ldots, s_k$ are the generators of $\mathcal{S}$ and

$$f_{s_i}(E) = \begin{cases} 0, & \text{if } [s_i,\, E] = 0 \\ 1, & \text{if } \{s_i, E\} = 0 \end{cases} \tag{1.20}$$

In order to perform error correction operations for a stabilizer code, we only need to measure the eigenvalues of the generators of the stabilizer group S, $(-1)^{f_{s_i}(E)}$ for each s_i. The error syndromes indicates which error has occurred, and correction operations can be applied correspondingly.

When logical information is encoded into a subsystem $\mathcal{L}$ of a subspace $\mathcal{C}$ in Hilbert space $\mathcal{H}$, we can write

$$\mathcal{H} = \mathcal{C} \oplus \mathcal{C}^{\perp} = (\mathcal{L} \otimes \mathcal{G}) \oplus \mathcal{C}^{\perp}, \tag{1.21}$$

where $\mathcal{G}$ is referred to as the gauge degrees of freedom in the subspace $\mathcal{C}$. Logical operators of the subsystem code are unitaries l that preserve the codespace, where undressed logical operators are of the form $l \otimes I$ and dressed logical operators are of the form $l \otimes g$ for some $g \in \mathcal{G}$. Using the stabilizer formalism, we can describe stabilizer subsystem codes by their gauge group $\mathcal{G}$. $\mathcal{Z}(\mathcal{G})$ is the stabilizer group S of the subsystem code, $\mathcal{N}(\mathcal{G})$ are the undressed logical operators, and $\mathcal{N}(\mathcal{S})$ are the dressed logical operators. For an $[\![n, k, d]\!]$ subsystem code, the distance d is the weight of the smallest operator in $\mathcal{N}(\mathcal{S})\backslash\mathcal{G}$. In the case that $\mathcal{G}$ is an abelian group, we would have $\mathcal{G} = \mathcal{S}$ and the resulting code is a stabilizer subspace code.

Given a subsystem code specified by the gauge group $\mathcal{G}$, new codes can be constructed by fixing gauge degrees of freedom in the system. By measuring a seleceted set of gauge operators $g \in \mathcal{G}$ and postselecting on outcome $+1$, a new gauge group $\mathcal{G}'$ with elements commuting with the measured gauge can be formed. This process is called gauge-fixing, it does not change the number of encoded qubits k and does not decrese the distance of the code d.

1.3.3 Quantum Fault-Tolerance and Thresholds

Given a quantum error correcting code of some distance, we would want to encode quantum information into this code in such a way that local errors smaller than some weight are correctable. Again assume errors aflicting our qubits are independent and identically

distributed with probability p for each fault location throughout all time steps of a quantum circuit. Our error correction protocol succeeds if we have correctly diagnosed the error that has occurred and faithfully restored our quantum information. The quantum fault-tolerant theorem states that when provided with imperfect devices of sufficiently low probability of failure, we can increase the size of encoding for our information and achieve computation with arbitrarily small logical failure rate [19]. When concatenated codes are considered, for sufficient accuracy arbitrarily long quantum computations can be implemented with polylogarithmic overhead. Similar statements has also been shown for topological codes.

When given an infinite code family $\{C_L\}$ parameterized by a growing size parameter L and an underlying decoder, we can define the logical error rate $p_{logical}(p, L)$ as the probability that the decoder for the size L code would fail at correcting an error that occurs at physical error rate p and introduce an unwanted logical operator. The accuracy threshold threshold p_{thr} is the largest physical error rate such that $\forall p \leq p_{thr}$,

$$\lim_{L\to\infty} p_{logical}(p, L) = 0. \tag{1.22}$$

Therefore, as long as $p \leq p_{thr}$ we can make our quantum computation arbitrarily accurate by increasing L. For a particular value of p and L, the pseudothreshold is defined as the error rate where $p_{logical}(p, L) = p$. For physical error rates below the pseudothreshold, encoding information in the target error correcting code leads to lower logical error rate than directly using an unencoded system.

1.3.4 Classical Error Correcting Codes

Due to the demands in communication networks, the theory of classical error correction is well-developed using classical information theory. The basic idea of classical error correction is to encode a bit into multiple bits, send the bit through a noisy communication channel, and decode the bit being received on the other end of the channel. When the er-

ror probability is sufficiently small on each bit of information being transmitted, the error probability for the encoded information can be reduced. However, it is not straightforward to adapt classical error correction protocols in the quantum setting. The main reason being that since qubits are subject to the non-cloning theorem, quantum information cannot be duplicated easily as classical information. Moreover, since arbitrary measurements of the qubits cannot be performed without collapsing the quantum state, the decoding procedure for error correction codes has to be adapted in order to maintain the quantum information while having the capability of correcting errors.

Similar to the idea of quantum error correction as explained above, classical error correction embeds information into vectors within vector spaces constructed over $\mathbb{F}_2$ where vectors encoded with information are sufficiently orthogonal to each other so that they remain distinguishable even in the presence of local errors.

A $[n, k, d]$ classical linear code C is defined as the null space of a parity check matrix $H \in \mathbb{F}^{(n-k)\times n}$, $C = \{x \in \mathbb{F}_2^n | Hx = 0\}$. The code distance d is the minimum weight of operators to map between two different codewords. Equivalently, the linear code C can also be defined as the row space of the generator matrix $G \in \mathbb{F}_n^k$, $G = Ker(H)$. Popular choices of classical linear codes include classical LDPC codes, which generally are constructed through randomly generating large regular bipartite graphs of certain degrees [20, 21]. We will expand more on classical LDPC codes in Chapter 6.

1.3.5 3-Bit Code and 3-Qubit Code

As a fundamental example of quantum error correction codes, here we consider the three-qubit code, which is the natural extension of classical three-bit repetition code. In the classical case when given a bit of information we can produce 3 copies of that bit, transmitting the 3 bits through a noisy channel, and then determining the value of the original bit via majority voting. This code is capable of correcting at most one bit flip error among the 3 bits being transmitted, and if the error probability per bit is less than 0.5 then the

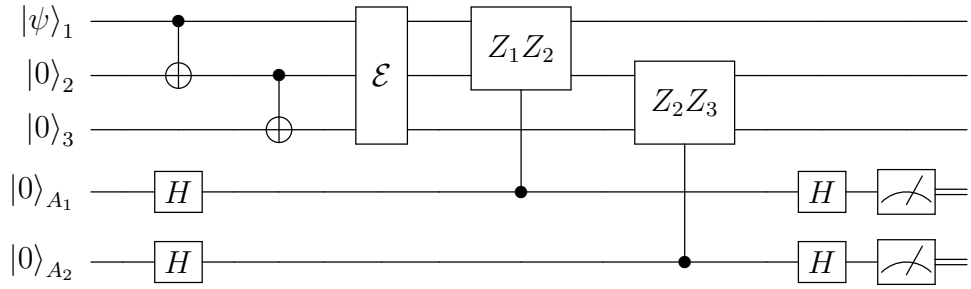


Figure 1.3: The quantum circuit for the three-qubit code. Logical state encoding: given qubit 1 in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, quantum operations are performed to entangle qubit 1 with qubit 2 and 3 both starting in the $|0\rangle$ state, creating a logical qubit in state $|\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle$. Error process: an error channel $\mathcal{E}$ is inserted to introduce errors into the circuit. Syndrome extraction: stabilizers Z_1Z_2 and Z_2Z_3 are measured via ancillary qubits A_1 and A_2 in order to determine the correction operation that should be applied.

Table 1.1: The syndrome for all bit-flip errors on the three qubit code. The syndrome is consisted of the measurement outcome on the two ancillary qubits A_1 and A_2. Notice that weight-2 and weight-3 errors in the right column have the same syndromes with weight-1 errors in the left column, showing that the 3-qubit code can only correct a weight-1 bit-flip error.

Weight-1 Error	Syndrome	Weight-2/3 Error	Syndrome
$I_1I_2I_3$	00	$X_1X_2X_3$	00
$X_1I_2I_3$	10	$I_1X_2X_3$	10
$I_1X_2I_3$	11	$X_1I_2X_3$	11
$I_1I_2X_3$	01	$X_1X_2I_3$	01

repetition code can reduce the error probability for the encoded information.

$$0 \to 000 \quad 1 \to 111 \tag{1.23}$$

In the quantum case when given a qubit in state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we can encode $|\psi\rangle$ into a three-qubit state $|\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle$ using the quantum circuit shown below. Since $|\psi\rangle_L$ is the eigenvalue 1 eigenvector of matrices Z_1z_2, z_1z_3 and z_2z_3, we say that $Z_1Z_2,\ Z_1Z_3,\ Z_2Z_3$ are the stabilizers of the state $|\psi\rangle_L$, and they form an abelian group with generators $G = \langle Z_1Z_2,\ Z_2Z_3\rangle$. By measuring the stabilizers of the code, we obtain "syndromes" which are information on whether the encoded information is corrupted,

and that can be used to find corresponding correction operations. Figure 1.1 shows the syndromes corresponding to different single qubit bit-flip errors. We can see that each single-qubit bit-flip error produces a unique syndrome S, therefore allowing us to correct each of these errors.

CHAPTER 2

EFFICIENT SAMPLING ALGORITHM FOR MONTE CARLO SIMULATIONS

This chapter contains results from

Muyuan Li, Mauricio Gutiérrez, Stanley E. David, Alonzo Hernandez, and Kenneth R. Brown, Phys. Rev. A. **96**, 032341 (2017) [22]

Due to the exponential scaling of classical variables required to track the time evolution of a full quantum state, classical simulation of large quantum systems is difficult. However, by using the Gottesman-Knill theorem, a subclass of quantum circuits can be efficiently simulated classically. These quantum circuits contain Clifford gates and Pauli measurements, and are often referred to as stabilizer circuits because they are restricted to transformations to and from elements within the Pauli group. For a majority of numerical results included in this thesis, we relied on Monte Carlo sampling of stabilizer circuits to obtain error rates for various quantum error-correcting codes.

For a given QEC circuit and a physical noise model, obtaining an exact algebraic expression for the logical error rate is in principle possible. In the context of a stabilizer code and a noise model consisting of stochastic discrete errors, this amounts to (1) enumerating every possible error configuration on the circuit, (2) calculating its probability of occurrence, and (3) determining whether or nor it results in a logical error. The logical error rate is then given by:

$$p_L = \sum_{i=0}^{N_c} A_i \, p_i, \tag{2.1}$$

where N_c is the total number of error configurations, A_i is the probability of occurrence of error configuration i, and $p_i = 0(1)$ if the error configuration i is correctable (uncorrectable). Although possible in principle, the exact computation of a logical error rate is infeasible in practice, due to the high cardinality of the error configuration set. For a circuit

with n_g gates and s possible different errors after each gate, $N_c = (s+1)^{n_g}$. Even a modest circuit, like a level-1 QEC routine for a distance-3 code consisting of 3 rounds of stabilizer measurements, contains more than 100 gates, making the exact computation of the logical error rate impractical.

It is, therefore, common to employ Monte Carlo methods to estimate p_L [23, 24, 25, 26]. The basic procedure to obtain a logical error rate consists of two steps: (1) the generation of a faulty circuit (an error configuration) based on the physical noise model and (2) the simulation of the circuit to determine if that particular error configuration is correctable. In this chapter, we describe the two different Monte Carlo methods used in the first step. The first one utilizes traditional direct sampling of the whole error configuration set. The second one relies on importance sampling of the error configuration subsets that are relevant to the logical error rate. Both methods are based on direct sampling, i.e., each error configuration is completely uncorrelated from the previous ones.

2.1 Traditional Sampler

To generate an error configuration, the traditional sampler traverses the circuit exhaustively and after each gate an error is inserted with a probability given by the noise model. This approach is convenient for high error rates. However, it is problematic if the error rate is low, because most of the times no error is inserted. For illustrative purposes, consider a simple case where the error rate $p = 0.1\%$ is the same for ech gate and the circuit has $N_g = 100$ gates. The traditional sampler will return an error-free circuit $(1-p)^{n_g} \approx 90\%$ of the runs. Furthermore, if the circuit corresponds to a fault-tolerant QEC protocol of a distance-3 code, by construction no error configuration of weight-1 will result in a logical error. This implies that $(1-p)^{n_g} + n_g p(1-p)^{n_g-1} \approx 99.5\%$ of the runs will generate an error configuration which is known a priori to be correctable. The limitation of the traditional sampler becomes even more dramatic for lower error rates and codes of higher distance [27].

2.2 Importance Sampler

It is possible to split the error configuration set into subsets based on each configuration's error weight (number of errors). The key advantage of the importance sampler relies on two features of this particular subset splitting: (1) it is straightforward to compute analytically the total probability of occurrence of each subset (the sum of the probabilities of occurrence of the error configurations in the subset) and (2) for low error rates we can safely ignore high weight error subsets, since their probability of occurrence will be vanishingly small.

Here we consider subsets in terms of both errors that occur after single-qubit gates and two-qubit gates. For example, a subset labelled by (s, t) contains all error configurations of the circuit with s errors after single-qubit gates and t errors after two-qubit gates. Notice that an error after a two-qubit gate can still be of weight 1. Let n_s and n_t be the total number of single-qubit gates and two-qubit gates in this circuit. We assume an error model where the error probability is the same for all single-qubit gates (p_s) and the error probability is also the same for all two-qubit gates (p_t). The probability of occurrence of the error subset (s, t) is then:

$$A_{s,t}(p_s, p_t) = \binom{n_s}{s} p_s^s (1 - p_s)^{n_s - s} \binom{n_t}{t} p_t^t (1 - p_t)^{n_t - t}. \tag{2.2}$$

Given a quantum circuit and a noise model, the algorithm to estimate the logical error rate using the importance sampler consists of three steps:

1. Select a tolerance value. This corresponds to the total added probability of occurrence of the high weight subsets that will not be sampled. In the worst scenario imaginable, every error configuration in the excluded subsets would result in a logical error. The tolerance value represents the worst-case discrepancy between the real and the obtained logical error rate. In particular, it provides a worst-case upper bound to the logical error rate.

2. For each error subset (s, t), use direct Monte Carlo sampling to approximate its logical error rate $p_L(s, t)$. The sampling is done by randomly selecting s single-qubit gates and t two-qubit gates and adding errors after them.

3. Calculate the total logical error rate for the circuit:

$$p_L(p_s, p_t) = \sum_{s,t} A_{s,t}(p_s, p_t) p_L(s, t), \tag{2.3}$$

where $A_{s,t}(p_s, p_t)$ is computed analytically using Equation 2.2.

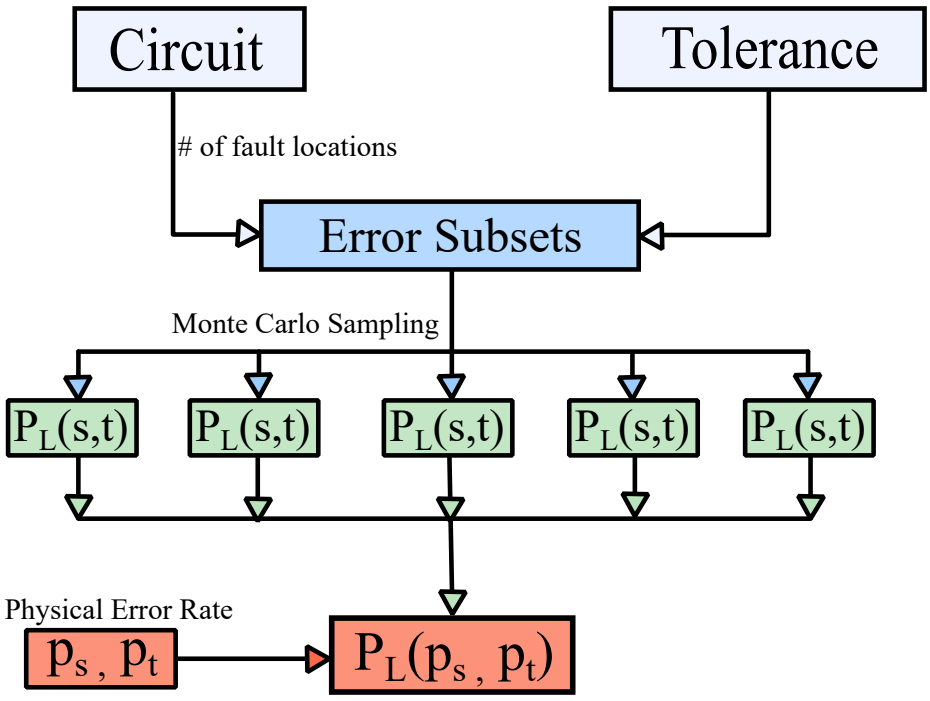


Figure 2.1: Block diagram for using importance sampling to compute the logical error rate of a quantum circuit. $P_L(s, t)$ in the green layer represents the logical error rate of the error subset (s, t) found through Monte Carlo sampling, and $P_L(p_s, p_t)$ in the red layer represents the final logical error rate of the entire circuit at physical error rates (p_s, p_t).

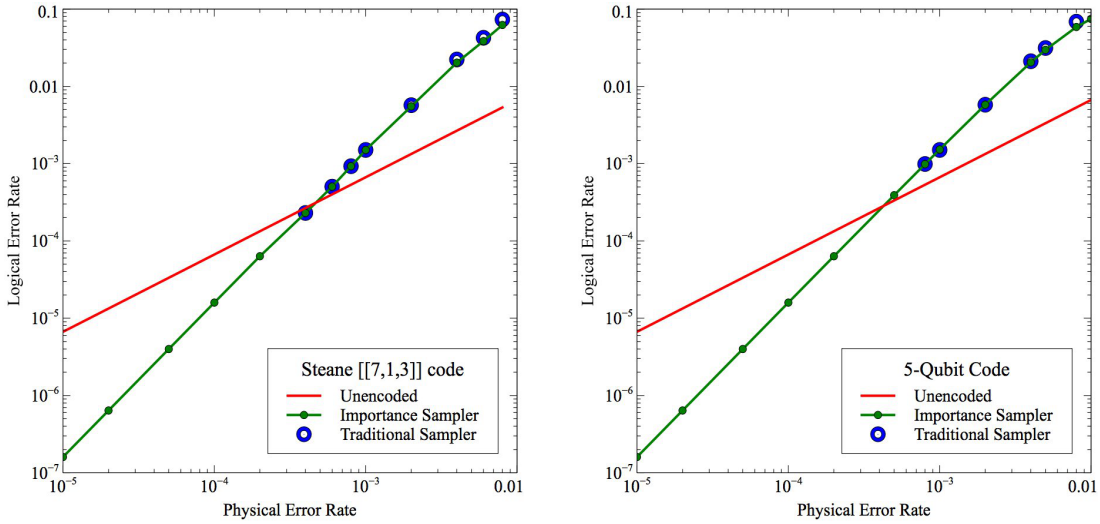
Notice that the logical error rate $p_L(s, t)$ for a particular (s, t) subset is not a function of the physical error rates p_s and p_t. This means that, for a given circuit and noise model, we can pre-compute the logical error rate for each relevant subset, and use those values to evaluate the total logical error rate for different physical error rates. In contrast to the tradi-

tional sampler, there is no need to re-run the Monte Carlo simulations for different values of p_s and p_t. We simply calculate new values for the probabilities of subset occurrence $A_{s,t}$ and compute p_L using Equation 2.3. Once the Monte Carlo simulations for each relevant subset are done, the logical error rates for different physical error rates can be computed at no additional cost, making the importance sampler much more efficient than the traditional one. However, for high physical error rates (about 10^{-2} and higher) the importance sampler becomes either inaccurate or very slow. The trade-off arises because as the error rate increases so does the probability of occurrence of higher weight subsets and therefore the number of subsets that need to be included to obtain an accurate result. In this error regime, it is suitable to employ the traditional sampler.

2.3 Comparing Performance of Sampling Algorithms

In order to test the performance of the importance sampler, we simulated the logical error rate for the Steane $[\![7, 1, 3]\!]$ code, the five-qubit code, and the Bare $[\![7, 1, 3]\!]$ code with both the traditional and the importance sampling algorithm. To make sure that the results of the two sampling algorithms are comparable, we analytically proved that the probability for an arbitrary error to occur at any location in the circuit is the same for both sampling algorithms.

From both Fig. 2.2 and Fig. 2.3 we can see that the results of the two sampling algorithms coincide exactly at lower physical error rates, but can start to diverge as physical error rate increases, with the logical error rate obtained from the traditional sampler slightly higher than that from the importance sampler. This is because for a given number of subsets, at lower physical error rates these subsets are enough to calculate the cumulative logical error rate to a high accuracy, but for higher physical error rates the importance sampler must sample a larger number of error subsets in order to achieve the same level of accuracy in the cumulative logical error rate. In this case, the traditional sampler accounts for all error subsets of this circuit, while the importance sampler only accounts for a small number of

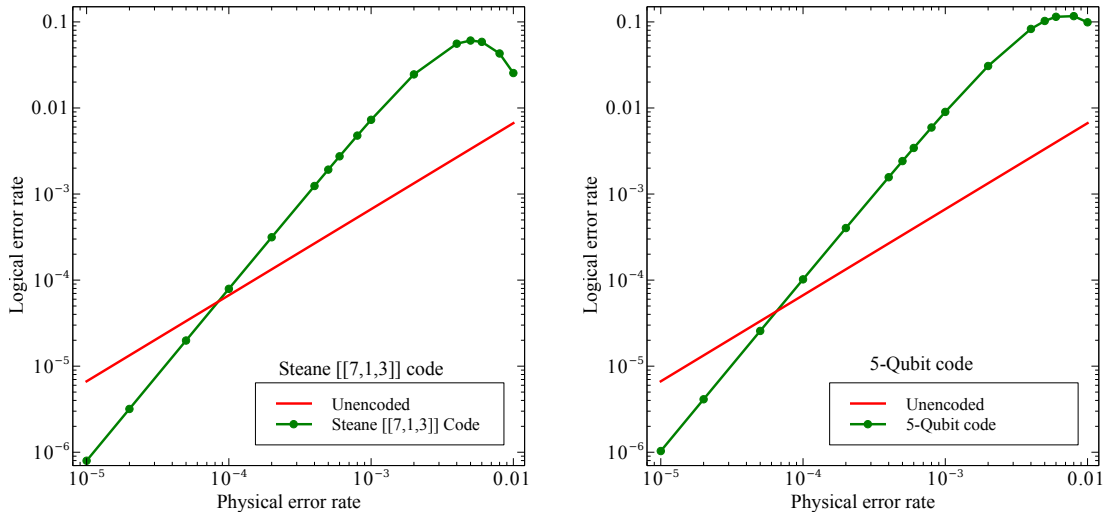


(a) Steane $[\![7, 1, 3]\!]$ code

(b) Five-qubit code

Figure 2.2: Logical error rate with the importance sampler and the traditional sampler for the Steane $[\![7, 1, 3]\!]$ code and the five-qubit code with Shor-style ancillary qubits under the standard depolarizing error model. Up to about $p = 2.0 \times 10^{-3}$, the two samplers result in essentially the same logical error rate.

subsets, resulting in a smaller logical error rate.



(a) Steane $[\![7, 1, 3]\!]$ code

(b) Five-qubit code

Figure 2.3: Logical error rate with the importance sampler for the Steane $[\![7, 1, 3]\!]$ code and the five qubit code with Shor-style ancillary qubits under the anisotropic error model. Around $p = 1.0 \times 10^{-2}$, the probability of occurrence of the high-weight errors becomes significant and ignoring them causes a dip on the logical error rate. The dip occurs at a lower p under the anisotropic error model because there are more faulty locations than under the standard model.

CHAPTER 3

FAULT TOLERANCE WITH BARE ANCILLARY QUBITS FOR A $[\![7, 1, 3]\!]$ CODE

This chapter contains results from

Muyuan Li, Mauricio Gutiérrez, Stanley E. David, Alonzo Hernandez, and Kenneth R. Brown, Phys. Rev. A. **96**, 032341 (2017) [22]

For a given QEC code, different methods of syndrome measurement, state preparation and decoding can all affect the resulting logical error rate. For syndrome measurement, using a single (bare) ancillary qubit to measure a stabilizer is in general not fault-tolerant, since errors on the ancillary qubit can propagate to the data and form uncorrectable errors [1, 28]. Although there are various methods to make the syndrome measurement fault-tolerant, these come at the expense of extra resources. Shor's method requires a w-qubit cat state to measure a weight-w stabilizer [29]. An extra qubit is needed to verify the ancillary qubit, but this is not a strict requirement [30, 31]. Steane's method requires the fault-tolerant preparation of a logical state [32], while Knill's method relies on the fault-tolerant preparation of a logical Bell pair [23].

Using bare ancillary qubits for stabilizer measurement can be fault-tolerant if we are guaranteed that single-qubit errors or errors that occur with a probability linearly proportional to the physical error rate do not propagate to form uncorrectable errors. More specifically, if we assume only Pauli errors, then after measuring a stabilizer of weight w a weight $\lfloor w/2 \rfloor$, where $\lfloor x \rfloor$ is the floor of x, error can propagate to the data. This can still be fault-tolerant in several cases. In some codes, the presence of gauge subsystems allow the decomposition of high weight stabilizers into lower weight gauge operators that can be measured in a fault-tolerant fashion using bare ancillary qubits. This is the case of the Bacon-Shor codes [33, 30]. In other cases, codes have a large enough distance such that any error of weight up to $\lfloor w/2 \rfloor$ can be corrected ($d \geq 2\lfloor w/2 \rfloor + 1$). This is the case of

large distance surface [25, 34] and color codes [35, 36]. Finally, certain codes with distance $d < 2\lfloor w/2 \rfloor + 1$ allow for stabilizer measurement with bare ancillary qubits because the resulting errors on data qubits are correctable. This is the case for the $[\![9, 1, 3]\!]$ surface code, where the weight-2 errors that propagate when measuring the weight-4 stabilizers are all correctable for specific orderings of the entangling two-qubit gates [31].

In this chaper, we present a non-CSS $[\![7, 1, 3]\!]$ QEC code that falls into the latter category: every single-qubit Pauli error on the ancillary qubit propagates to the data qubits to form a correctable error. The code was found by a numerical greedy search for stabilizer codes where single-qubit errors on the ancillary qubits do not lead to a logical error [37]. We refer to this code as the Bare $[\![7, 1, 3]\!]$ code, and we refer to the syndrome measurement method of one ancillary qubit per stabilizer as the bare method.

3.1 Details of the Bare $[\![7, 1, 3]\!]$ Code

The Bare $[\![7, 1, 3]\!]$ code was found through a numerical greedy search of stabilizer codes with the property that single-qubit errors on the bare ancillary qubit would not lead to a logical error. Table 3.1 presents the stabilizer generators and logical X and Z operators of the code. As seen from the stabilizers, the code is non-CSS and degenerate.

Table 3.1: List of stabilizers and logical operators X_L, Z_L for the Bare $[\![7, 1, 3]\!]$ code.

Stabilizer Generators	Logical Operators
X_0X_4	
X_1X_4	
X_2X_5	$X_L = X_1X_2X_3$
X_3X_6	$Z_L = Z_0Z_1Z_4$
$Z_2Z_3Y_5Y_6$	
$Z_0Z_1Z_2X_3Z_4Z_5$	

FIG 3.1 shows the configuration of the stabilizers when the Bare $[\![7, 1, 3]\!]$ code is embeded in a plane.

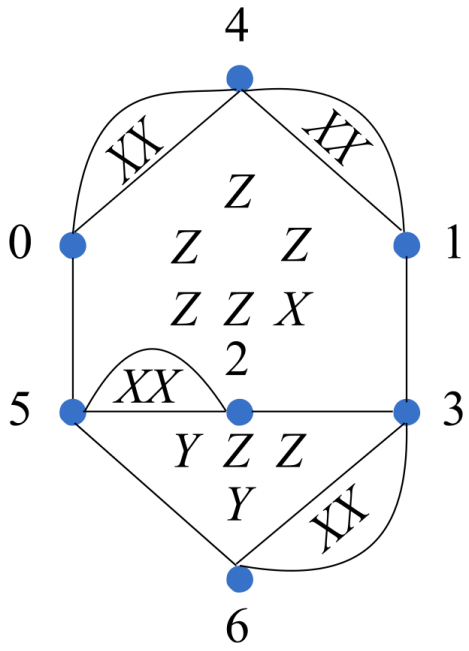


Figure 3.1: The Bare $[\![7, 1, 3]\!]$ code embeded in a plane. Each vertex represents a data qubit, and each stabilizer generator of the code corresponds to a face in this graph.

3.1.1 Single-qubit Errors on a Bare Ancillary Qubit

To illustrate how using bare ancillary qubits can lead to uncorrectable errors in some QEC codes, let us consider the $[\![7, 1, 3]\!]$ Steane code. From Figure 3.2, we can see that when measuring the stabilizer $X_3X_4X_5X_6$, a single qubit X error on the bare ancillary qubit propagates to form the error X_5X_6 on the data qubits. Although this error is detectable at a

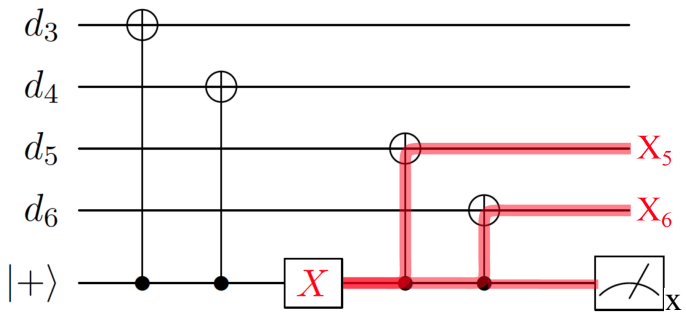


Figure 3.2: For the Steane $[\![7, 1, 3]\!]$ code with bare ancillary qubits, a single-qubit error on the ancillary qubit would lead to a logical error.

later stage, its syndrome is equivalent to an X_0 error. When this correction is applied, the resulting error $X_0X_5X_6$ is equivalent to the logical operator $X_L = X_0X_1X_2X_3X_4X_5X_6$ up to a stabilizer.

On the other hand, in the Bare $[\![7, 1, 3]\!]$ code, all single-qubit errors on the ancillary qubits propagate to become errors with unique syndromes, hence correctable. This is achieved by considering different schedules for coupling the data to the ancillary qubit when measuring the stabilizers. Note that all ancillary qubits are prepared in the $|+\rangle$ state.

Table 3.2: Single-qubit error syndromes for the Bare $[\![7, 1, 3]\!]$ code.

$Z_0 \to 100000$	$X_0 \to 000001$	$Y_0 \to 100001$
$Z_1 \to 010000$	$X_1 \to 000001$	$Y_1 \to 010001$
$Z_2 \to 001000$	$X_2 \to 000011$	$Y_2 \to 001011$
$Z_3 \to 000101$	$X_3 \to 000010$	$Y_3 \to 000111$
$Z_4 \to 110000$	$X_4 \to 000001$	$Y_4 \to 110001$
$Z_5 \to 001010$	$X_5 \to 000011$	$Y_5 \to 001001$
$Z_6 \to 000110$	$X_6 \to 000010$	$Y_6 \to 000100$

The syndromes for the 21 single-qubit Pauli errors are shown in Table 3.2. Notice that the syndromes of Z_2Z_3 is 001101 which is distinct from all syndromes of single-qubit errors. Similarly, $Z_0Z_2 \to 101000$, $Z_0Z_2X_3 \to 101010$, and $Z_4Z_5 \to 111010$ all have unique syndromes. Since each gate used in a measurement acts between an ancillary qubit and a data qubit where the ancillary qubit controls a Pauli operator on the data qubit, this observation suggests the following syndrome measurement coupling schedule:

1. For each weight-2 generator, the measurement gates can be coupled to the ancillary qubit in any order.

2. For the stabilizer $Z_2Z_3Y_5Y_6$, couple the measurement gates in left-to-right order.

3. For the stabilizer $Z_0Z_1Z_2X_3Z_4Z_5$, couple the measurement gates in order of Z_0, Z_2, X_3, Z_1, Z_4, Z_5.

By using this coupling schedule, for the stabilizer $Z_2Z_3Y_5Y_6$, a single-qubit error on the

ancillary qubit can propagate the error Z_2Z_3 onto the data qubits, which has a unique syndrome; for the stabilizer $Z_0Z_1Z_2X_3Z_4Z_5$, a single-qubit error on the ancillary qubit can propagate errors Z_0Z_2, $Z_0Z_2X_3$, and Z_4Z_5 onto the data qubits, which all have unique syndromes. Therefore, all single-qubit errors on the ancillary qubits propagate to exclusively correctable errors.

3.1.2 Two-qubit Errors on a Bare Ancillary Qubit

The Bare $[\![7, 1, 3]\!]$ code is vulnerable to "hook errors". Figure 3.3a illustrates an instance of a 2-qubit Pauli error that can occur under the standard symmetric depolarizing error model. This XX error propagates to become $Z_1X_2X_3Z_4Z_5$, which has syndrome 101011, the same syndrome as $Y_1Z_4Z_5$. Thus, the correction $Y_1Z_4Z_5$ will be applied, and the resulting error is $X_1X_2X_3$, the logical X operator.

In Fig.3.3b the 2-qubit XX error on the control-Y gate propagates to become the error X_5Y_6, which has syndrome 000111, the same as Y_3. After the correction Y_3 is applied, the resulting error $Y_3X_5Y_6$ is equivalent to the logical Z operator up to a stabilizer.

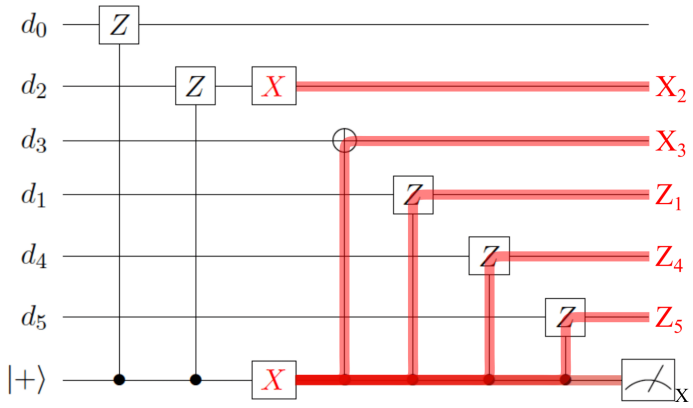
3.2 Error Models

3.2.1 Error Models

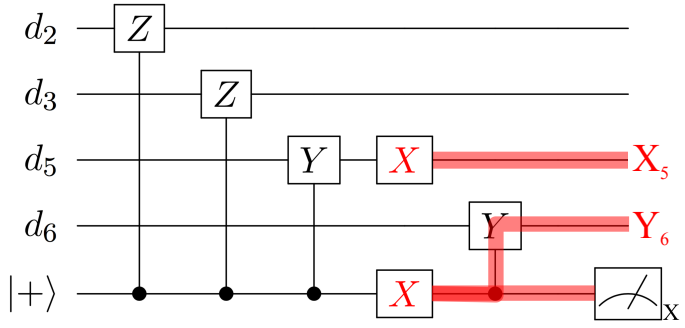
To study the properties of the Bare $[\![7, 1, 3]\!]$ code, we applied two Pauli error models: the standard depolarizing error model and an anisotropic error model.

Standard Depolarizing Error Model

The standard depolarizing error model is a common quantum channel where, after each gate, there is a symmetric depolarization with some probability p. After single-qubit gates (including Pauli gates and ancillary qubit preparation gates), a traceless single-qubit Pauli operator, randomly selected from $P = \{X, Y, Z\}$, is applied with probability p_s. After two-qubit gates, a traceless two-qubit Pauli operator, randomly selected from $Q =$



(a)



(b)

Figure 3.3: Two examples of 2-qubit XX errors that propagate to become uncorrectable errors. In (a) the error leads to an X_L, while in (b) the error results in a Z_L.

$\{I, X, Y, Z\} \otimes \{I, X, Y, Z\} \setminus \{II\}$, is applied with probablity p_t. Although in general $p_s \neq p_t$, it is common for them to be simulated with the same value [23]. A measurement reports the wrong result with probability p_{meas}. For the cases studied here, p_{meas} is set to the single qubit error probability p_s.

Anisotropic Error Model

For single qubit gates, preparation, and measurement errors remain the same as the standard depolarizing error model. The conceptual motivation for the anisotropic error model is that two-qubit gate errors occur due to errors in the gate coupling. As an example, consider errors in the entangling Mølmer-Sørensen (MS) gate used in trapped ions [38]. The MS gate corresponds to $\exp(-i(\pi/4)XX)$, and an over or under-rortation could result in

the coherent error $\exp(-i(\epsilon/2)XX)$, where ϵ denotes the over- or under-rotation angle. For ϵ randomly and symmetrically distributed around 0, this error can be described as a random application of the Pauli error XX with probability p_t, which is determined by the distribution of θ.

The key idea is that the two-qubit correlation in the error is not any two-qubit error, but only the two-qubit Pauli error aligned with the gate. The error after a Control-P gate is then ZP with probability p_t. For the specific case of ion traps, this can be derived by including the single qubit rotations necessary to transform the MS gate into a Control-P gate. In this study, we do not include the individual rotations but work at the level of Control-P gates. We expect random errors on the individual qubits in addition to the control error from the two-qubit gate. To account for the random errors, after every two-qubit gate we apply the two-qubit anisotropic error with probability p_t followed by single qubit errors with probability p_s on the two qubits involved in the gate.

3.2.2 Fault-tolerance Dependent on Error Model

When performing the bare method under the two error models defined in Section 3.2.1, we can see that all two-qubit errors in the anisotropic error model can be detected and corrected, because a Z error on the control qubit does not propagate to form other errors on the data. However, for certain two-qubit errors in the standard depolarizing error model, such as the XX error as seen in Fig. 3.3a and Fig. 3.3b, the errors propagate to form uncorrectable hook errors. Therefore, the Bare $[\![7, 1, 3]\!]$ code with the bare method can achieve fault-tolerance under the anisotropic error model because all errors that occur with probability linear in the error rate of physical operations are correctable; however, the same syndrome measurement method cannot achieve fault-tolerance under the standard depolarizing error model since certain two-qubit errors that occur with probability linear in the physical error rate propagate to become uncorrectable logical errors.

3.3 Fault-tolerance With a Flag Qubit

Since the bare method is not fault-tolerant for the Bare $[\![7,1,3]\!]$ code under the standard depolarizing error model, additional resources would be required to ensure fault-tolerance. In this section we propose the method of using two additional flag qubits to perform fault-tolerant syndrome measurement on the Bare $[\![7,1,3]\!]$ code. The idea of the flag method was used by Yoder and Kim [39], and presented in detail by Chao and Reichardt [40].

Since the source of logical errors for the Bare $[\![7,1,3]\!]$ code are the hook errors propagated from errors on the ancillary qubit, we can use flag qubits to detect errors on the ancillary qubit that can propagate to form these hooks. In order to use the flag method, we need to change the order of the gates to measure the stabilizer $Z_0Z_1Z_2X_3Z_4Z_5$ to: $Z_0, X_3, Z_4, Z_2, Z_1, Z_5$. The purpose of this order change is to ensure that all errors that could trigger the flag qubit measurement have distinct syndromes. As we can see in TABLE 3.4, this change of order makes the Bare $[\![7,1,3]\!]$ code vulnerable to certain single-qubit errors that were correctable with the old order. But since these errors all trigger the flag qubit measurement and all result in distinct syndromes, they are still correctable.

The circuits to fault-tolerantly measure the high-weight stabilizers are shown in Figure 3.4. The second ancillary qubit initialized in the $|0\rangle$ state in both circuits is the flag qubit. With no errors present, these circuits behave exactly the same as the circuits using bare ancillary qubits for syndrome measurement, and the Z-basis measurements on the flag qubits will always give $|0\rangle$. In Figure 3.4a errors after gate a and d cannot create uncorrectable hook errors, and in Figure 3.4b errors after gate a and f cannot create uncorrectable hook errors. Errors after all other gates in these two circuits will be detected by causing a $|1\rangle$ measurement outcome for the flag qubit. These errors and their syndromes are listed in Table 3.3 and Table 3.4. Each of these errors has a distinct syndrome, so it can be corrected. Note that Y errors on the ancilla qubit have the same effect on the data as X errors.

As argued by Chao and Reichardt [40], this method of syndrome measurement is fault-

tolerant. Under the standard depolarizing model, all errors that appear with probability linear in the physical error rate can be detected and corrected using the above flag method, hence making the Bare $[\![7,1,3]\!]$ code fault-tolerant under the standard depolarizing error model.

Table 3.3: The nontrivial data errors that can result from a single two-qubit error for the stabilizer $Z_2Z_3Y_5Y_6$ (Figure 3.4a) and trigger the flag qubit measurement. The errors marked in red can lead to logical errors with the bare method.

b errors	Data error	syndrome	c errors	Data error	syndrome
IX	Y_5Y_6	001101	IX	Y_6	000100
XX	$X_3Y_5Y_6$	001111	XX	X_5Y_6	000111
YX	$Y_3Y_5Y_6$	001010	YX	Y_5Y_6	001101
ZX	$Z_3Y_5Y_6$	001000	ZX	Z_5Y_6	001110

Table 3.4: The nontrivial data errors that can result from a single two-qubit error for the stabilizer $Z_0Z_1Z_2X_3Z_4Z_5$ (Figure 3.4b) and trigger the flag qubit measurement. The errors marked in red can lead to logical errors with the bare method.

b errors	Data error	syndrome	c errors	Data error	syndrome
IX	$Z_1Z_2Z_4Z_5$	100010	IX	$Z_1Z_2Z_5$	010010
XX	$Z_1Z_2X_3Z_4Z_5$	100000	XX	$Z_1Z_2X_4Z_5$	010011
YX	$Z_1Z_2Y_3Z_4Z_5$	100101	YX	$Z_1Z_2Y_4Z_5$	100011
ZX	$Z_1Z_2Z_3Z_4Z_5$	100111	ZX	$Z_1Z_2Z_4Z_5$	100010
d errors	Data error	syndrome	e errors	Data error	syndrome
IX	Z_1Z_5	011010	IX	Z_5	001010
XX	$Z_1X_2Z_5$	011001	XX	X_1Z_5	001011
YX	$Z_1Y_2Z_5$	010001	YX	Y_1Z_5	011011
ZX	$Z_1Z_2Z_5$	010010	ZX	Z_1Z_5	011010

3.4 Simulation Scheme

In this section we present the circuit used in the simulation, the QEC scheme used to perform correction after each round of the Bare $[\![7,1,3]\!]$ code with noise, and the calculations

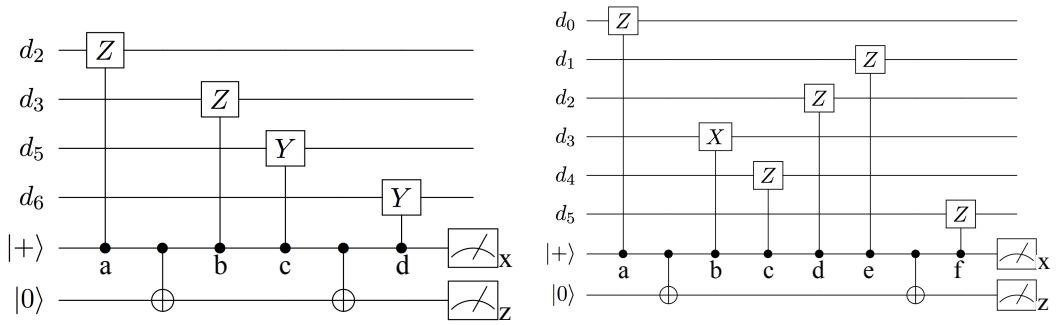


Figure 3.4: Flagged syndrome measurement for the Bare $[\![7, 1, 3]\!]$ code. (a) Circuit to fault-tolerantly measure the syndrome of the stabilizer $Z_2Z_3Y_5Y_6$ using a flag qubit. (b) Circuit to fault-tolerantly measure the syndrome of the stabilizer $Z_0Z_1Z_2X_3Z_4Z_5$ using a flag qubit. Notice the order of gates is different from what we used in the the bare method.

we performed to obtain logical error rates. The simulations were done using the CHP stabilizer simulator [41]. The simulation follows the 1-rectangle (1-Rec) formalism [19] for distance-3 and level-1 encoding, where the 1-Ga we are simulating is the level-1 logical identity gate.

3.4.1 Simulation Circuit

Fig. 3.5 shows the quantum circuit we constructed to simulate one error-correcting step of the bare method for the Bare $[\![7, 1, 3]\!]$ code. The simulated gate is a logical identity followed by an error-correction gadget. The gates are grouped together for each ancillary qubit measurement, and the order of qubit-coupling between the data qubits and ancillary qubits follows exactly the scheduling as required for fault-tolerant measurements of each stabilizer generators. For each gate shown in this circuit, errors of appropriate type are inserted after it with a probability given by the noise model. The circuit for simulating the flag method requires two additional flag qubits, and has a different order of qubit-coupling between the data and ancillary qubits, as described in Section 3.3.

The syndrome measurement is repeated up to three times with freshly prepared ancillary qubits. The repetition is necessary due to errors during or between syndrome measurements

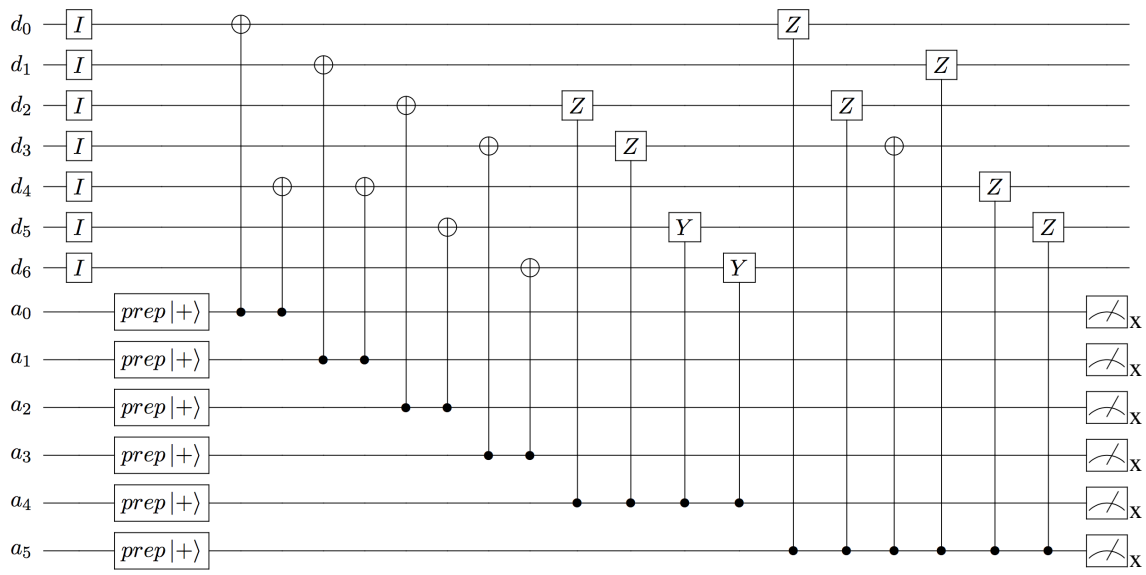


Figure 3.5: The circuit simulating the Bare $[\![7, 1, 3]\!]$ code and its measurement steps. The circuit is constructed with only Clifford gates, and errors (single-qubit and two-qubit) are inserted after ancillary qubit preparation, CNOT, Control-Z, Control-Y, and measurements.

[42]. If the first two error syndromes agree, correction is performed based on the error syndrome. If the two syndromes disagree, a third syndrome is measured and correction is performed based on the third syndrome.

3.4.2 Logical Error Rate Calculation

We calculate the logical error rate under the two different error models for various physical error strengths. For each run of in the Monte Carlo simulation, we initialize all the data qubits in state $|0\rangle$, 6 of the ancillary qubits in state $|+\rangle$ for syndrome measurements, and 2 of the ancillary qubits in state $|0\rangle$ to be used as flag qubits for the high-weight stabilizers. We then perform one round of noise-free stabilizer measurements to project the state of the data qubits to the logical $|0\rangle$ state. Then the simulation proceeds as follows:

1. Perform 2 or 3 rounds of error correction with random errors inserted using the importance sampling scheme.

2. Apply the decoder to determine the corresponding error configuration and correct

accordingly.

3. Perform noise-free correction to the final state to project the state back to the codespace.

4. If the final state is different from the initial one, count as one logical error.

In the importance sampling scheme (see Appendix A.2), we classify the error configurations into subsets according to the number of single- and two-qubit errors present in the configuration. The subset consisting of configurations of s single-qubit errors and t two-qubit errors is labelled by (s, t). For example, if during the execution of the QEC circuit, two single-qubit preparations were faulty and an error occurred after one of the CNOTs, this error configuration belongs to the subset $(2, 1)$.

We perform Monte Carlo simulations on selected error subsets and compute the logical error rate per subset, $p_L(s, t)$, by calculating the ratio of the successful runs over the total number of runs. The total logical error rate, p_L, is calculated as a weighted average over the selected error subsets:

$$p_L(p_s, p_t) = \sum_{s,t} A_{s,t}(p_s, p_t) p_L(s, t),$$

where $A_{s,t}$ is the probability of occurrence of subset (s, t), that is, the total probability of occurrence of all error configurations with s single-qubit errors and t two-qubit errors.

We use the calculated logical error rates at different physical error strengths to estimate the pseudothreshold of the code under a particular error model [43]. The pseudothreshold is the intersection between the physical error rate line $y = 2p/3$ and the logical error rate $p_L(p_s, p_t)$. The reason for using this line instead of $y = p$ is that, if we assume a symmetric depolarizing noise model on a single qubit, the infidelity is given by $2p/3$. More intuitively, if we focus on a single-qubit Pauli state, like $|0\rangle$, only two Pauli errors (X and Y in this case) will cause an error.

3.5 Results

In this section we present and analyze the results from numerical Monte Carlo simulations of the Bare $[\![7, 1, 3]\!]$ code under different noise models using the two different syndrome measurement methods. We first focus on the standard depolarizing error model and then on the anisotropic error model. We show that using the bare method, the Bare $[\![7, 1, 3]\!]$ code does not have a pseudothreshold under the standard depolarizing error model, due to some uncorrectable two-qubit errors that have probability linear in the physical error strength.

3.5.1 Standard Depolarizing Error Model

Fig. 3.6 presents the logical error rates for the standard depolarizing model with $p_s = p_t = p$ for several values of p. For the bare method, the best fit corresponds to a function with linear term as leading term, while for the flag method the best fit corresponds to a function with quadratic term as leading term.

For the bare method, although there is an intersection between the logical error rate and the physical error rate, the curves still remain parallel for physical error rates below the intersection. The fact that there is still a linear term in the logical error rate implies that we would not observe an exponential suppression on the logical error rate with subsequent levels of concatenation. Thus, this intersection is not a pseudothreshold for the Bare $[\![7, 1, 3]\!]$ code, and the code is not fault-tolerant under the standard depolarizing error model. Examples of why this would happen are shown in Fig. 3.3.

For the flag method, all errors that occur with probability linearly proportional to the physical error rate can be corrected. By performing a quadratic fit for the logical error rates and computing the intersection of the fitting curve with the error rate of an unencoded qubit, we obtain a level-1 pseudothreshold of 1.08×10^{-3}.

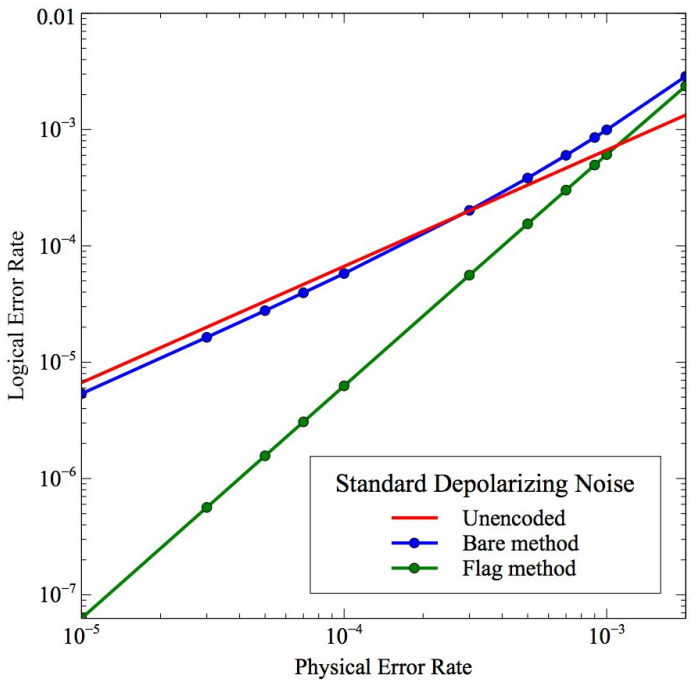


Figure 3.6: Logical error rate for the Bare $[\![7, 1, 3]\!]$ code under the standard error model. For the bare method, the logical error rate of the code remains parallel to the physical line for physical error rates below the intersection. This implies that the logical error rate is linear in p. For the flag method, we observe a level-1 pseudothreshold of 1.08×10^{-3}.

3.5.2 Anisotropic Error Model

We now calculate the pseudothreshold of the Bare $[\![7, 1, 3]\!]$ code under the anisotropic error model. In this model, the physical error strength after each gate is p, and $p_s = p_t = p$.

By performing a quadratic fit for the logical error rates and computing the intersection of the fitting curve with the error rates of an unencoded qubit, we obtain level-1 pseudothreshold for the Bare $[\![7, 1, 3]\!]$ code with both syndrome measurement methods. This is because all single-qubit and two-qubit errors that occur with probability linear in p are guaranteed to be detected and corrected by the Bare $[\![7, 1, 3]\!]$ code. From Fig. 3.7 we can see that for the bare method, the level-1 pseudothreshold is at 2.0×10^{-4}. For the flag method, the level-1 pseudothreshold is at 4.0×10^{-4}.

The main reason for the difference between the two pseudothresholds is that the number of uncorrectable error configurations with probability quadratically proportional to the

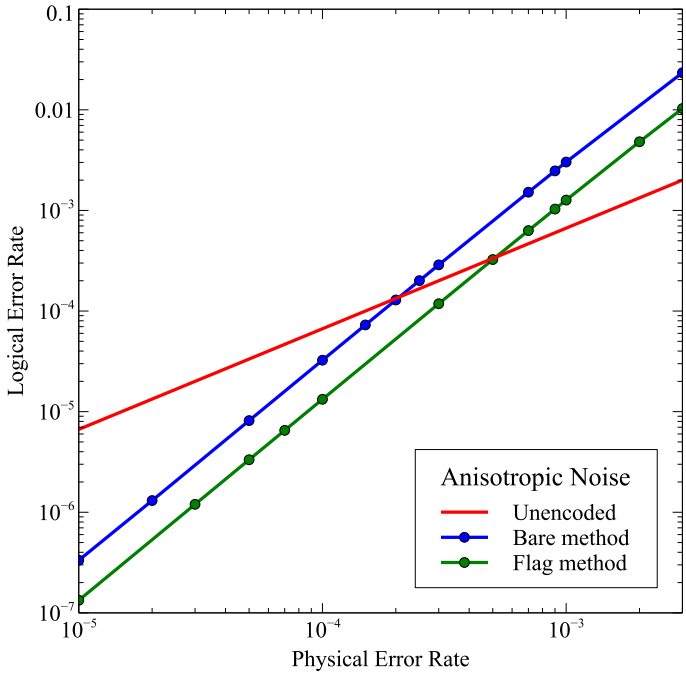


Figure 3.7: Logical error rate for the Bare $[\![7, 1, 3]\!]$ code under the anisotropic error model, $p_s = p_t$. For the bare method, the level-1 pseudothreshold is at $p = 2.0 \times 10^{-4}$. For the flag method, the level-1 pseudothreshold is at $p = 4.0 \times 10^{-4}$. The Bare $[\![7, 1, 3]\!]$ code is fault-tolerant under the anisotropic error model with both syndrome measurement methods.

physical error rate is higher for the bare method. Consider, for example, that some errors with probability quadratically proportional to the physical error rate can behave the same as the two-qubit errors listed in Table 3.3 and Table 3.4. Those errors could lead to logical errors with the bare method, but can be corrected with the flag method.

3.5.3 Comparison of Results across Error Models

From the results presented in Fig. 3.6 and Fig. 3.7 we can see that the level-1 pseudothreshold under the anisotropic error model is lower than that under the standard depolarizing error model. This is because in the anisotropic error model, after each two-qubit gate we are simulating a single-qubit random Pauli error on each qubit with probability p_s (Section 3.2.1), leading to more possible error locations in the overall circuit. If we assume the single-qubit errors and two-qubit errors are equally probable, as we did in the results shown

in Fig. 3.6 and Fig. 3.7, errors after the two-qubit gates are 3 times more likely to occur under the anisotropic error model than under the standard depolarizing model, because of the extra single-qubit Pauli errors. Hence, the Bare $[\![7,1,3]\!]$ code under the anisotropic error model has a lower pseudothreshold than under the standard depolarizing model.

3.6 Conclusions

We have presented properties of a new $[\![7,1,3]\!]$ stabilizer code that can achieve fault-tolerant syndrome measurements using a single ancillary qubit under the anisotropic error model, while fails to do so for the depolarizing model. In particular, the limit on total number of syndrome outcomes makes it impossible for the lookup table decoder to detect and correct all two-qubit errors in the standard depolarizing error model. Additionally, we showed that using the flag method the Bare $[\![7,1,3]\!]$ code can achieve fault-tolerance under both the standard depolarizing model and the anisotropic model. Overall, with the flag method the Bare $[\![7,1,3]\!]$ code shows better performance under realistic error models.

We note that the $[\![7,1,3]\!]$ triangle code presented by Yoder and Kim [39] also does not require additional ancillary qubit preparations and is able to achieve fault-tolerance under the standard depolarizing error model. However, their code would not be identified by the search criteria that identified our code [37]: a syndrome measurement with strictly bare ancillary qubits on their code would not be able to correct all hook errors originated from a single qubit error on the ancillary qubit. Instead they proposed an interwoven syndrome measurement method to achieve fault-tolerance. Since the interwoven method is not applicable for the Bare $[\![7,1,3]\!]$ code, no direct comparison between the two codes can be easily made. But if the flag method [40] was to be applied on both codes, they can both achieve fault-tolerance under the standard depolarizing error model with the same amount of resources, and the pseudothreshold we obtained for the Bare $[\![7,1,3]\!]$ code for the 1-Rec is comparable to the pseudothreshold reported by Yoder and Kim for the exREC [19].

CHAPTER 4

DIRECT MEASUREMENT OF BACON-SHOR STABILIZERS

This chapter contains results from

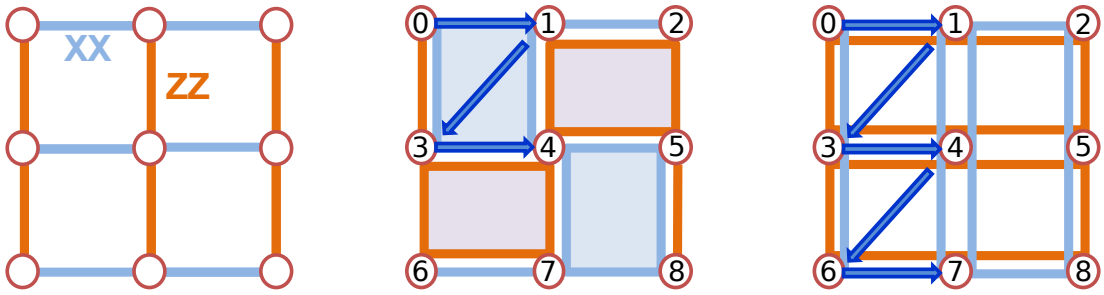
Muyuan Li, Daniel Miller, and Kenneth R. Brown, Phys. Rev. A. **98**, 050301 (**R**) (2018) [44]

The Bacon-Shor code is a subsystem code and has no threshold as L grows [45] but promising performance for small distance codes with a pseudothreshold of 0.2% for a depolarizing error model [30] and a fault-tolerant protocol for implementing universal gates without distillation [46]. The rotated surface code has L^2-1 check operators with $(L-1)^2$ checks of weight 4 in the bulk and $2(L-1)$ checks of weight 2 on the boundary [47]. The advantage of the Bacon-Shor code comes from using $2L(L-1)$ weight-2 gauge operators to determine the weight-$2L$ check operators and the lack of threshold is a result of having only $2(L-1)$ checks [33, 30, 45].

In this chapter, we first describe the relationship between the Bacon-Shor and surface codes with the compass model. We then use the gauge operators defined by the compass model to generate circuits that fault-tolerantly measure the weight-$2L$ Bacon-Shor stabilizers and the weight-4 surface code stabilizers using bare ancillary qubits. We then compare state preparation for the $L \times L$ Bacon-Shor and surface codes. Finally, we examine how these general results lead to a practical advantage for the Bacon-Shor code over the surface code on a 3×3 lattice under a generic error model and a more specific ion trap error model.

4.1 Compass Model and Error Correcting Codes

In condensed matter physics the compass model is used to describe a family of lattice models involving interacting quantum degrees of freedom [48]. The relationship between compass model and topological quantum computing is also well-studied, e.g, Kitaev's hon-



(a) Compass Model (b) Surface (c) Bacon-Shor

Figure 4.1: The compass model with ZZ bonds along the vertical axis and XX bonds along the horizontal axis. Choices of gauge on a 3×3 lattice lead to two well-known stabilizer codes: $[\![9, 1, 3]\!]$ surface code and $[\![9, 1, 3]\!]$ Bacon-Shor code . The underlying bonds of the compass model are a guide for how to fault tolerantly measure surface code and Bacon-Shor code stabilizers with bare ancillary qubits. Measuring stabilizers in order of gauge operators can help suppress hook errors on two-qubit gates in the stabilizer measurement circuit. The blue arrows show the circuit order for measuring an X-type stabilizer for both codes.

eycomb model [49]. Fig. 4.1a shows a 9-qubit compass model on a square lattice. Subsystem error-correcting codes arise naturally from the compass model, where the interactions between neighboring qubits can be viewed as weight-2 gauge operators [33]. The compass model also has a $90°$ rotation symmetry, so that X and Z Pauli errors are treated symmetrically. Examining Fig. 4.1 we see that the stabilizer generators of the rotated surface code and the Bacon-Shor code can be considered as different constraints on the parity of products of gauge operators. This process of choosing constraints is referred to as gauge fixing. For the surface code, each weight-2 stabilizer is exactly a gauge operator of the corresponding type, and each weight-4 stabilizer is equivalent to fixing the parity of the product of two gauge operators on the same face; for the Bacon-Shor code, each weight-L stabilizer is equivalent to fixing the parity of the product of L gauge operators in the same double row or column of qubits.

A challenge for constructing fault-tolerant syndrome measurement circuits is the propagation of errors from the ancillary qubits used for measurement to the data qubits that hold the information. This type of error is commonly referred to as a hook error, a generalization of the definition in Ref. [50]. A k-fault-tolerant procedure yields the correct result if k or

less errors happen. For a quantum error correcting code of distance d, the code is $\lfloor d/2 \rfloor$-fault-tolerant to single qubit errors when syndrome measurements are perfect. In order to maintain a $\lfloor d/2 \rfloor$-fault-tolerant procedure for noisy syndrome circuits, the syndromes must be measured multiple times and the measurement circuit must not generate uncontrolled errors on data qubits. The original approach of fault-tolerant syndrome measurement uses verified multi-qubit ancillary states to ensure each faulty gate in the syndrome circuit could produce only one error on the data [29, 32, 23]. A second approach allows multiple errors to propagate to the data, but uses ancillary qubit measurements to inform the decoder of these errors [51, 39, 40]. The third approach notes that some multi-qubit errors can be tolerated as if they were single-qubit errors [31].

The surface code is considered a promising candidate for fault-tolerant quantum computing [52, 25, 53]. It is also a popular choice for implementing error correction on near-term small quantum devices [34, 31, 54, 55] due to its ability to restrict all stabilizer measurements as local operations. Fault-tolerant syndrome measurement with a bare ancillary qubit per check operator relies on a proper choice of circuit for implementing syndrome measurement, as illustrated in Fig.4.1b. This choice has been previously described as directing the hook errors away from the direction of the logical error [31], but when viewed from the perspective of the compass code the measurement circuits are determined by the underlying gauge operators.

The two-dimensional Bacon-Shor code has $L-1$ X stabilizers that correspond to a double column of Xs and the $L-1$ Z stabilizers that correspond to a double row of Zs. The logical X operators are $X^{\otimes L}$ acting on all qubits in the same column, and logical Z operators are $Z^{\otimes L}$ acting on all qubits in the same row. Bacon-Shor is a subsystem code and the extra degrees of freedom result in gauge operators acting trivially on the encoded information [30]. We use these gauge degrees of freedom to fault-tolerantly and directly measure each stabilizer using a single bare ancillary qubit.

4.2 Fault-tolerance with the Bacon-Shor Code

The challenge of fault-tolerant $2L$-weight stabilizer measurement is that errors on the ancillary qubit can generate hook errors of weight $\lfloor (L-1)/2 \rfloor$ on the data. By measuring the stabilizers following the structure of the gauge operators, the hook errors are simply products of gauge operators and a single qubit error. Therefore, hook errors are equivalent to single qubit errors and $\lfloor d/2 \rfloor$-fault-tolerance is preserved for the code. Topologically one can consider the sequential product of gauge errors as a string with both ends attached to the same boundary and is therefore a trivial operator. This is in contrast to logical operators where the strings connect opposite boundaries [56].

For the X stabilizers, the circuit consists of preparing the ancillary qubits in $|+\rangle = X|+\rangle$ and then performing $2L$ controlled-not gates with each data qubit as the target and the ancillary qubit as the control, followed by measurement in the X basis. We order the controlled-not gates such that the target qubits come in pairs that follows the XX gauge operators. The stabilizer $S_{i,x} = \prod_{j=1,L} X_{j,i}X_{j,i+1}$, where $X_{j,i}X_{j,i+1}$ is the gauge operator on the qubits in row j and columns i and $i+1$, yields the sequence of controlled-nots $\prod_{j=1,L} \mathrm{CNOT}(a_i, d_{i,j})\mathrm{CNOT}(a_i, d_{i+1,j})$ where the target qubits d are paired by the gauge operator and controlled by the ancilla a. (Fig. 4.1c). A similar order holds for the Z stabilizers, where now the ancilla is prepared and measured in the Z basis and the controlled-nots target the ancilla qubits. This circuit has already been implemented experimentally in trapped ions and superconducting qubits for the $L = 2$ Bacon-Shor quantum error detection code [57, 58].

In addition to syndrome measurement, preparation of logical states is also an essential part of quantum error correction. For the surface code, to encode logical $|0\rangle$ we prepare all the data qubits in the physical $|0\rangle$ state, measure X type stabilizers L times and perform correction based on the syndrome. This procedure measures $(L^2-1)/2$ X stabilizers L times and requires $2L^2(L-1)$ two-qubit gates. For Bacon-Shor, to encode logical $|0\rangle$ we

prepare L L-qubit GHZ states in the X basis along the rows without verification. Generating the L GHZ states only requires $L(L-1)$ two-qubit gates, significantly fewer gates than preparation by projecting onto the stabilizers. We note that this deterministic preparation is fault-tolerant for Bacon-Shor codes, since each row fails independently and $\lceil L/2 \rceil$ rows must fail for the preparation to yield an incorrect logical measurement. Like the Bacon-Shor code itself, this preparation has no threshold and for fixed qubit error rate, there will be an optimal L for state preparation.

4.3 Surface-17 VS Bacon-Shor-13

We now consider the smallest Bacon-Shor code on the 3×3 lattice. We need one ancillary qubit per stabilizer to perform fault-tolerant syndrome measurement of the $[\![9, 1, 3]\!]$ Bacon-Shor code, yielding a total of 13 qubits. We refer to this choice as Bacon-Shor-13 following the notation of Tomita and Svore where the $[\![9, 1, 3]\!]$ surface code with 8 ancilla is referred to as Surface-17 [31].

To compare the performance of Bacon-Shor-13 and Surface-17 we focus on simulating a circuit with 3 elements: logical state encoding, quantum error correction, and measurement of the individual data qubits. We refer to this circuit as the simple circuit. Using the measurement results, we determine the outcome of the logical circuit and the probability that the circuit fails.

To perform error correction using Surface-17 and Bacon-Shor-13, we designed a two-step lookup table decoders for both codes. The details of the Surface-17 decoder can be found at [55]. For any two-step decoder, in the first step if the syndrome shows no errors then no correction is performed; if the syndrome shows errors, then a second syndrome is measured and correction is applied based on the second syndrome. All simulations are performed using the stabilizer simulation program CHP [41] inside an importance sampling routine [22, 55]. State preparation of Bacon-Shor determinstically constructs the logical state, $|0\rangle_L = \bigotimes_{i=0}^{2} \frac{1}{\sqrt{2}}(|+++\rangle + |---\rangle)_{0+i,3+i,6+i}$ while Surface-17 requires 2 or 3

rounds of X stabilizer measurement due to the inherent projection noise of the measurements [55].

4.3.1 Error Models

We consider two error models: a standard depolarizing error model and an ion trap inspired error model. We assume arbitrarily connected qubits for both cases, but for the ion trap error model the duration of two-qubit gates depends on ion distance [55]. Here we describe the error model in terms of Kraus operators.

Standard Error Model: The one- and two-qubit Kraus channels are of the form

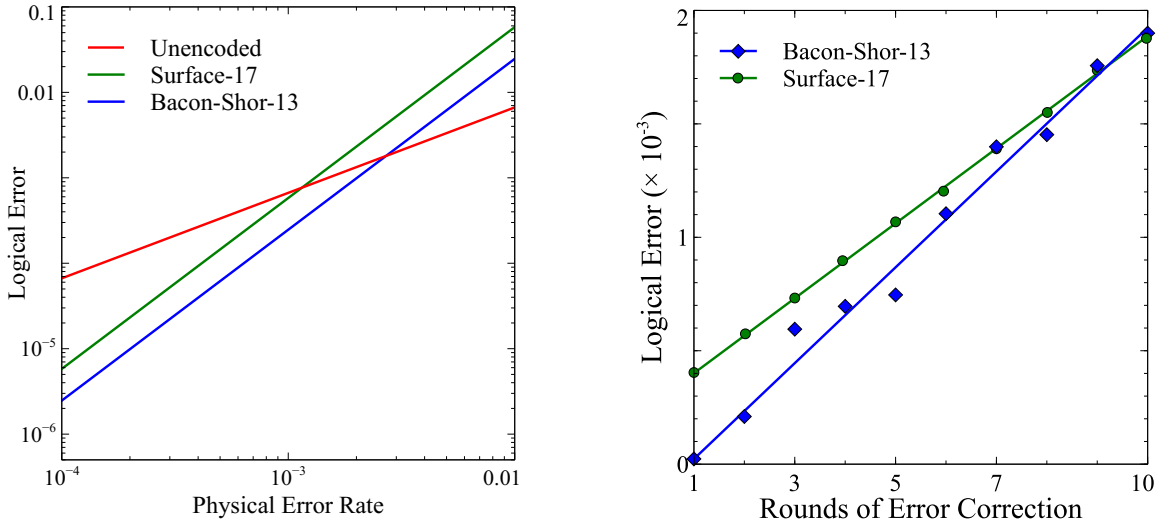
$$\begin{aligned} E_1 &= \{\sqrt{1-p}I, \sqrt{\frac{p}{3}}X, \sqrt{\frac{p}{3}}Y, \sqrt{\frac{p}{3}}Z\}, \\ E_2 &= \{\sqrt{1-p}II, \sqrt{\frac{p}{15}}IX, \ldots, \sqrt{\frac{p}{15}}ZZ\}, \end{aligned} \tag{4.1}$$

where p is the error rate of the error channel. For each gate in the circuit, an element from the one-qubit (two-qubit) Pauli group is sampled and applied after the gate (before for measurements) to serve as an error.

Ion Trap Error Model: Quantum error correction circuits are constructed from Mølmer-Sørensen entangling gates, $\exp(-i(\theta/2)XX)$, and single qubit rotations, ***RX*** and ***RY*** [59, 60, 55]. The error model assumes that the main source of error comes from fluctuations in the control strength. The Kraus channels for the control errors associated with these gates are

$$\begin{aligned} E_{XX} &= \{\sqrt{1-p_{XX}}II, \sqrt{p_{XX}}XX\}, \\ E_{RX} &= \{\sqrt{1-p_{RX}}I, \sqrt{p_{RX}}X\}, \\ E_{RY} &= \{\sqrt{1-p_{RY}}I, \sqrt{p_{RY}}Y\}, \end{aligned} \tag{4.2}$$

where in our simulation p_{XX} is the Mølmer-Sørensen control error rate and $P_{XX} = 10\,p_{RX} = 10\,p_{RY}$. The Mølmer-Sørensen gate couples ions through the collective ion motion and an error can occur due to heating of the motion during the gate. We can add a motional mode



(a) One round of error correction

(b) Multiple rounds of error correction

Figure 4.2: Comparison of Bacon-Shor-13 and Surface-17 in a simple circuit simulation. (a) With one round of error correction Bacon-Shor-13 shows a pseudothreshold of 0.9% and Surface-17 shows a pseudothreshold of 0.1%. The difference is mainly due to the difference in logical state preparation. (b) At a physical error rate of 10^{-3}, Surface-17 starts to outperform Bacon-Shor-13 in more than 9 rounds of error correction.

heating error with the Kraus channel

$$E_{heating} = \{\sqrt{1-p_h}II, \sqrt{p_h}XX\}, \tag{4.3}$$

where $p_h = \dot{\bar{n}}/2k \times t_{MS}$, where $\dot{\bar{n}}$ is the heating rate, k is the number of phase space loops of the motion during the gate, and t_{MS} is the duration of the corresponding entangling Mølmer-Sørensen gate. This error model follows Ref. [61].

During one and two-qubit gates, uncontrolled Stark shifts can also lead to a gate-induced dephasing error modeled with the single-qubit Kraus channel

$$E_{dephasing} = \{\sqrt{1-p_d}I, \sqrt{p_d}Z\}, \tag{4.4}$$

where $p_d = r_d \times t_g$, r_d is the dephasing rate and t_g is the time of the applied gate.

4.4 Results

The results of the standard depolarizing error model are shown in Fig.4.2. The logical error of the simple circuit scales quadratically with the physical error rate showing that both protocols are 1-fault-tolerant as expected (Fig.4.2a). The higher logical error of Surface-17 is due primarily to the overhead of state preparation. When multiple rounds of error correction are executed, Surface-17's advantage in error correction starts to dominate. As shown in Fig. 4.2b, Surface-17 starts to outperform Bacon-Shor-13 in a simple circuit with more than 9 rounds of error correction when the physical error rate is 10^{-3}. Each round of error correction allows for a logical gate and the performance of 9 rounds of error correction should be compared to the error of performing 9 physical gates $\approx 9 \times 10^{-3}$. The Bacon-Shor-13 logical error shows more variation around the trend line for odd rounds of error correction. We suspect this is due to the subsystem nature of the code but have not found a clear reason.

To simulate the performance of error-correcting codes in trapped-ion system we map the codes onto a linear ion chain and compile controlled-not gates from Mølmer-Sørensen gates [38, 60]. The linear chain can be split apart to allow measurement of ancillary qubits without affecting the data and then rejoined for further quantum operations Ref. [55]. Using a simulated annealing algorithm, we searched for ion chain arrangements that minimized total time for quan- tum error correction or average two-qubit gate time. In our ion trap error model gate error scales with two-qubit gate time and minimizing average two-qubit gate time minimizes the error due to ion heating and dephasing. Times required to execute the simple circuit when the ions are arranged to minimize the average two-qubit gate time are shown in TABLE 4.1. Bacon-Shor-13 has shorter circuit execution time. These reported times are not fundamental and can be improved by changes in gate and measurement schemes [62, 6] but for all protocols, Bacon-Shor-13 will maintain the time advantage.

Table 4.1: Time required (in μs) to execute a simple circuit in an ion trap model (see [55] and Supplementary Material). The times can vary as the rounds of stabilizer measurements depends on the results of syndrome measurements.

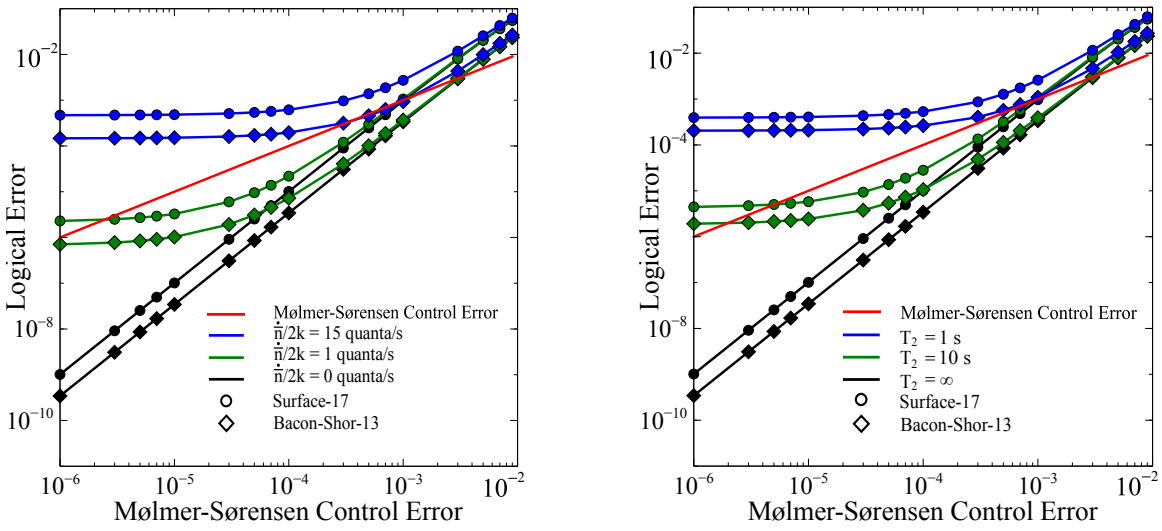
Code	Prep	QEC	Measure	Total
Surface-17	2400-3600	4900-9800	100	7400-13500
Bacon-Shor-13	1670	4310-8620	100	6080-10390

In Fig. 4.3 we present the results of comparing Bacon-Shor-13 and Surface-17 in a simple circuit with one round of error correction under the influence of different ion trap error sources. Here we assume the error rate for single-qubit gates and measurements is $\frac{1}{10}$ of two-qubit gates in order to match realistic error rates in experiments. All results are computed for an optimization of ion chain arrangement that minimizes average 2-qubit gate times for both codes independently.

From Fig. 4.3 we notice that for all error sources and strengths Bacon-Shor-13 outperforms Surface-17. For both codes, the simulated simple circuit would outperform a single two-qubit gate assuming only Mølmer-Sørensen control errors with rate below 10^{-3}. The heating rate and gate-induced dephasing error both raise the effective two-qubit error rate and this further separates the performance between Bacon-Shor-13 and Surface-17 for this circuit. These results indicate that a total two-qubit gate error rate of 10^{-3} should clearly demonstrate that an encoded circuit outperforms the unencoded circuit.

4.5 Ion Trap Operation Times

Here we present some results on trap operation times for ion arrangements optimized according to different parameters. More details of the ion trap model and justification for the parameters are available in Ref. [55]. The optimization takes into account that although 2-qubit gates can occur between any pair of ions, the gate time will depend on the ion distance. It also assumes that before the ancillary qubits are measured they must be physically displaced from the data qubits. If the data qubits and ancillary qubits are separated in the chain, the number of joining and splitting steps is reduced at the cost of longer logical gate



(a) Ion heating error

(b) Spin dephasing error

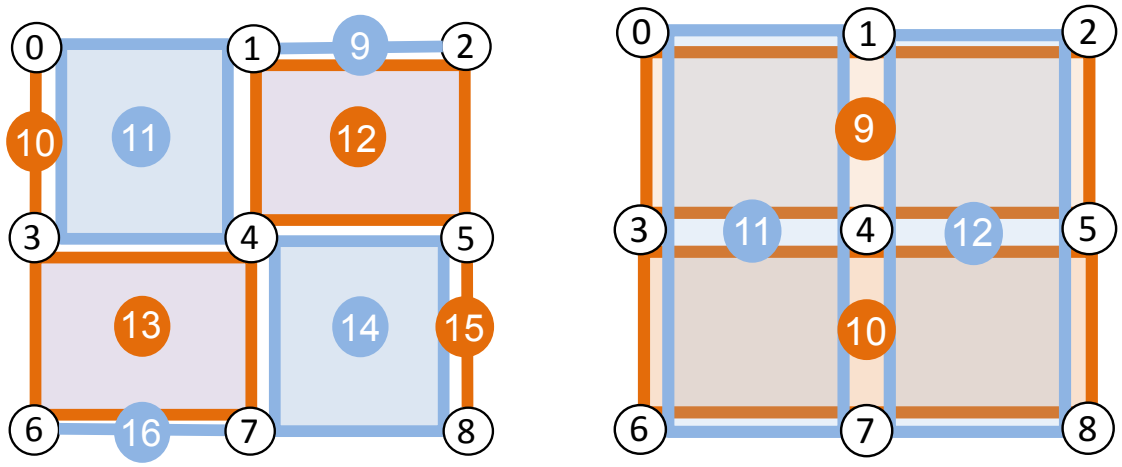
Figure 4.3: Comparison of Bacon-Shor-13 and Surface-17 with a simple circuit under the influence of an ion trap error model (see text). Logical error is plotted as a function of a control error in the Mølmer-Sørensen gate. Each plot includes an additional error due to ion heating ($\dot{\bar{n}}$) or gate-induced dephasing (T_2) that depends on gate time.

Table 4.2: Ion arrangements optimized for an array of parameters (see text). Numbers in bold face represent ancillary qubits.

Code	Opt.	Ion Arrangement																
Surface-17	SA	0	2	6	8	1	4	3	7	5	**11**	**12**	**10**	**15**	**13**	**14**	**9**	**16**
	MA	2	**9**	1	**12**	5	**15**	8	**14**	4	**11**	0	**10**	3	**13**	7	**16**	6
	MT	**10**	**15**	**9**	5	0	1	**11**	**12**	**14**	7	4	3	8	2	6	**13**	**16**
Bacon-Shor-13	SA			0	2	6	8	1	3	7	5	4	**11**	**10**	**12**	**9**		
	MA			8	2	**12**	1	5	**9**	4	**10**	7	3	**11**	0	6		
	MT			2	1	5	4	**9**	**12**	**10**	**11**	7	3	0	8	6		

times. We note that neither limit is fundamental. It is possible to make 2-qubit gates where there is no time dependence on ion distance by adjusting laser power [62] and that ancillary qubits do not need to be separated when two ion species are used [6].

The optimization was done using a simulated annealing algorithm with an objective function adjusted for different optimization parameters, total time assuming two parallel gates (T) or average 2-qubit gate time (A), and different constraints, mixed (M) or separated (S) data and ancillary qubits. For each optimization label we calculate times for operations performed in both serial and parallel. Parallel operations allow for two simul-



(a) Surface-17 (b) Bacon-Shor-13

Figure 4.4: Qubit labeling for the $[\![9, 1, 3]\!]$ surface code and Bacon-Shor code. Data qubits, ancillary qubits for X type stabilizers, and ancillary qubits for Z type stabilizers are shown in black, blue, and orange circles, respectively.

taneous two-qubit gates exciting the independent x and y radial modes and fully parallel single-ion operations. We assume single-qubit gates, parallel measurement/state preparation, and shuttling between operation and measurement zones require 10 μs, 100 μs, and 100 μs, respectively. The shuttling operation includes the time to split and join ion chains. The results in the chapter are for the MA optimization, which minimizes error for the error correction step. In Ref. [55], the reported results for Surface-17 are for MT.

Table 4.3: Trap operation times (in μs) for one round of error correction calculated according to ion arrangements optimized for an array of parameters (see text). All values are reported in μs and the numbers in parentheses refer to the gate time when two 2-qubit operations are performed in parallel.

Code	Opt.	Logic	Shuttle	Meas.	Total
Surface-17	SA	7240 (3920)	200	100	7950 (4220)
	MA	2300 (1170)	1800	800	4900 (3770)
	MT	4300 (2320)	700	300	5300 (3320)
Bacon-Shor-13	SA	5580 (3270)	200	100	5880 (3570)
	MA	2910 (1490)	1000	400	4310 (2890)
	MT	3580 (1860)	400	100	4080 (2360)

Table 4.4: Trap operation times (in μs) for logical state preparation calculated according to ion arrangements optimized for an array of parameters (see text). The time for Surface-17 can vary as it is a probabilistic circuit of syndrome extraction.

Code	Opt.	Prep Time
Surface-17	SA	7880-11820 (4460-6690)
	MA	2400-3600 (1200-1800)
	MT	3800-5700 (2100-3150)
Bacon-Shor-13	SA	670 (450)
	MA	1670 (1040)
	MT	1480 (970)

4.6 Conclusion

We have found that for subsystem and subspace codes derived from the compass model the gauge operators prescribe a method for fault-tolerant stabilizer measurement. This yields the ability to fault-tolerantly measure large stabilizer operators with single ancillary qubits and provides insight into previous work on small surface codes. These results does require increasing qubit connectivity and becomes impractical for local architectures as the stabilizer size grows. We have also noted that for any size Bacon-Shor code state preparation can be performed in a deterministic fault tolerant manner compared to the project and error correct method commonly proposed for state preparation.

These general results have practical implications for near term quantum devices. Specifically we have shown that Bacon-Shor-13 outperforms Surface-17 in all measures for a simple circuit: time, logical error rate, and number of qubits. The key advantage of Bacon-Shor-13 over Surface-17 comes from its greatly simplified state preparation. In addition, the lower qubit count makes Bacon-Shor-13 a more immediate target for near-term quantum error correction in systems where non-nearest neighbor gates are possible, such as trapped ions. However as Surface-17 holds advantage over Bacon-Shor-13 in terms of error correction, multiple rounds of error correction will begin to favor Surface-17.

We note that the compass model code on an $L \times L$ lattice allows for a family of codes

defined by how the gauge operators are fixed and the Bacon-Shor and rotated surface code are two extremes of this family. We are currently studying this family of codes to determine what choice of gauges would yield a threshold, as the codes transition from Bacon-Shor to surface code [63].

CHAPTER 5

LOGICAL PERFORMANCE OF 9 QUBIT COMPASS CODES IN ION TRAPS WITH CROSSTALK ERRORS

This chapter contains results from

Dripto M. Debroy, **Muyuan Li**, Shilin Huang, and Kenneth R. Brown, Quantum Sci. Technol. https://doi.org/10.1088/2058-9565/ab7e80 (2020) [64]

Quantum computing experiments have already demonstrated state stabilization [65, 66, 67, 68], single-axis quantum error correction [69, 70, 71, 72, 73, 74], multi-axis fault-tolerant quantum error detection [57, 58, 75], and we expect to be implementing full quantum error correction soon [76, 55, 77, 6]. These small quantum devices will be the predecessors to far larger fault tolerant quantum computers which can run interesting algorithms at high rates of success [78, 79, 80, 81].

One of the first important uses of these new devices will be to better understand the actual errors they face [82, 83]. This information will be used to find optimally performing codes and decoders for the error model of a given architecture, leading to improved logical performance. With this goal in mind, we study the performance of four $[\![9, 1, 3]\!]$ quantum error-correcting codes under a set of error models that are common to ion trap quantum computing systems. The codes being considered here are the 17-qubit rotated surface code [31], the 13-qubit Bacon-Shor code [33, 30, 44], and two variants of Shor's code [78]. There are many small QEC codes such as the $[\![5, 1, 3]\!]$ code [84, 85], Steane $[\![7, 1, 3]\!]$ code [86], Bare $[\![7, 1, 3]\!]$ code [22], twisted surface code [39], and tailored codes for biased error [87] that can be implemented using 10-20 qubits, with pseudothresholds that have been improved by the introduction of flag qubits [40, 88, 89, 90]. Here we picked our set of codes to be gauge fixes within the 2-D quantum compass code model [63]. As a result these codes require only bare ancilla for fault-tolerance and have high circuit-level

pseudothresholds.

While general quantum error correction literature considers the depolarizing error model [23, 52], in reality errors emerging in quantum systems are expected to be more architecture dependent. Hence when studying the performance of algorithms and error correcting schemes in realistic systems we have to take into account the errors that are dominant in the given architecture. We consider an ion trap quantum computer that defines the qubit using hyperfine clock states of $^{171}\mathrm{Yb}^{+}$ [91]. In this case $T_1 > 10^{10}s$, so we can ignore its effects over the course of an experiment. Although single-qubit gates in similar systems have been shown to have fidelity beyond the error-correction pseudothreshold of these small QEC codes [92, 93], there are a couple of limiting factors for two-qubit operation fidelities and qubit lifetimes in general that are native to the ion trap system. The most common sources of error to consider in a trapped-ion system are T_2 dephasing errors, motional mode heating errors in the trap, and overrotation and crosstalk errors induced by the application of gates via lasers [94].

Of these sources of noise, T_2 and overrotation noise are both shared among most qubit implementations, however the actual model for crosstalk noise is very architecture specific. As progress has been made towards increasing qubit count and improving control, these unwanted qubit-qubit interactions known as crosstalk errors have become a significant error source in near-term quantum devices [95]. In a trapped ion system, laser intensity spillover onto the neighboring ions during gate applications can lead to unwanted XX-type crosstalk errors between qubits involved in the desired gate and their neighbors in the ion chain. In a system implementing a small quantum error correcting code, such crosstalk errors can break fault tolerance and directly give rise to logical errors on the encoded information unless carefully dealt with. Therefore, in near-term quantum error correction experiments, steps must be taken to mitigate the damaging effects of crosstalk errors. Here we show that using a dynamic programming algorithm we can find optimal qubit to ion mappings for certain QEC codes that suppress the most damaging effects of first order crosstalk errors.

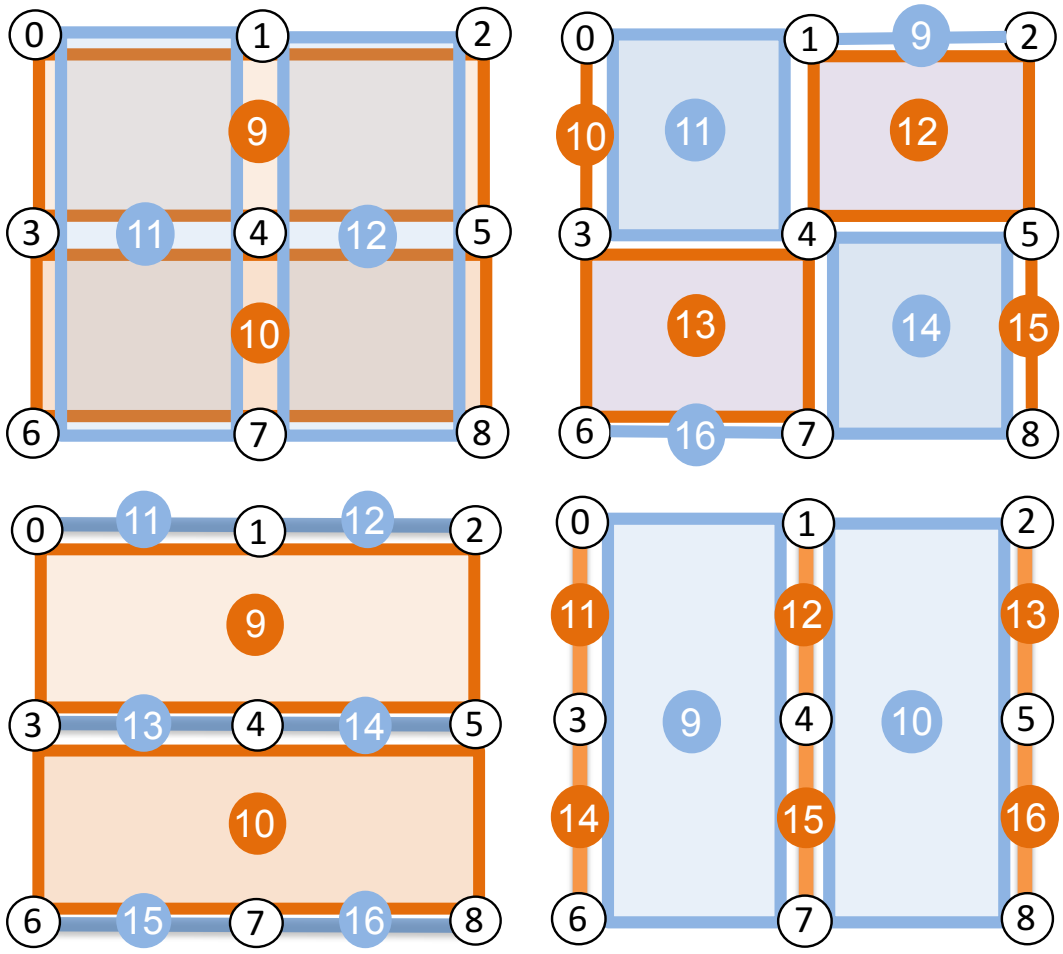


Figure 5.1: Stabilizer diagrams for (top row, left to right) Bacon-Shor-13, Surface-17, (bottom row, left to right) Shor-6X2Z, and Shor-6Z2X. The orange connections/plaquettes represent Z-type stabilizers and the blue connections/plaquettes represent X-type stabilizers. White circles represent data qubits and colored circles represent ancillary qubits which measure the stabilizer they are attached to.

Previous experimental work using trapped ions has already demonstrated implementation of classical error correction [96, 71], fault-tolerant quantum error detection [57] and logical state encoding of quantum error correcting codes [97, 58]. Several theoretical studies have examined possibilities of implementing quantum error correcting codes in near-term experiments using trapped ions, including architectural studies of connecting multiple traps via ion shuttling or optical interconnects [98, 99, 100, 101, 102, 5], while others have looked at logical performances of small error correction codes in realistic error models [55, 44, 103, 6, 104, 105].

In this chapter we study the logical performance of a transversal CNOT gate between two logical qubits that are maintained in the same trap under realistic error models featuring the mixing of overrotation, T_2 dephasing, and crosstalk. We present regions in which each code would perform the best in a near term experiment, along with the regions where the encoded qubit outperforms its physical counterpart. Our hope is that by finding optimal codes for these varied error models we will be able to hasten the arrival of successful error correcting implementations, which would be a major milestone in the pursuit of fault-tolerant quantum computing.

5.1 Error Correction in a Linear Ion Trap

5.1.1 9 Qubit Compass Codes

The quantum compass model on a square lattice of spins can be defined with the following Hamiltonian

$$H = \sum_{i} \sum_{j \neq L-1} J_X X_{i,j} X_{i,j+1} + \sum_{i \neq L-1} \sum_{j} J_Z Z_{i,j} Z_{i+1,j},$$

where ZZ type interactions occur on spins linked by a vertical edge, and XX type interactions for those sharing a horizontal edge [63]. Quantum error-correcting codes can be defined using the method of gauge fixing: inserting sets of these two-qubit gauge operators into the stabilizer group by fixing the eigenvalue of their products. These codes have a number of nice features: all of their stabilizers can be measured fault-tolerantly using bare ancillas, they can be modified to deal with spatially asymmetric noise, and they are easily decoded. In this chapter we consider 4 different $[\![9, 1, 3]\!]$ quantum error-correcting codes that can be defined using the compass model defined by a 3×3 square lattice: the rotated 17-qubit surface code (Surface-17) [31], the Bacon-Shor code (Bacon-Shor-13) [33, 22], and two variations of Shor's code (Shor-6Z2X, Shor-6X2Z) [78]. To clarify the code orientations we use, the stabilizers and logical operations of these codes are listed in Table 5.1. All of these codes can be implemented on a linear ion chain using at most 17 qubits to

Table 5.1: Stabilizers and logical operators of Bacon-Shor-13, Shor-6Z2X, and Shor-6X2Z, and Surface-17.

Bacon-Shor-13	Shor-6Z2X	Shor-6X2Z	Surface-17
Stabilizers			
$Z_0Z_3Z_1Z_4Z_2Z_5$	$X_0X_1X_3X_4X_6X_7$	$Z_0Z_3Z_1Z_4Z_2Z_5$	$Z_1Z_2Z_4Z_5$
$Z_3Z_6Z_4Z_7Z_5Z_8$	$X_1X_2X_4X_5X_7X_8$	$Z_3Z_6Z_4Z_7Z_5Z_8$	Z_0Z_3
$X_0X_1X_3X_4X_6X_7$	Z_0Z_3	X_0X_1	$Z_3Z_4Z_6Z_7$
$X_1X_2X_4X_5X_7X_8$	Z_1Z_4	X_1X_2	Z_5Z_8
	Z_2Z_5	X_3X_4	$X_0X_1X_3X_4$
	Z_3Z_6	X_4X_5	X_6X_7
	Z_4Z_7	X_6X_7	$X_4X_5X_7X_8$
	Z_5Z_8	X_7X_8	X_1X_2
Logical Operators			
$Z_0Z_1Z_2$	$Z_0Z_1Z_2$	$Z_0Z_1Z_2$	$Z_0Z_4Z_8$
$X_0X_3X_6$	$X_0X_3X_6$	$X_0X_3X_6$	$X_2X_4X_6$

protect one logical qubit of quantum information.

5.1.2 Gate Implementations

In this work we start with Clifford circuits composed of the gate set $\{X, H, CNOT\}$ along with preparation into the $|0\rangle$ state and measurement in the Z basis. It should be noted that since we do not allow preparation into $|+\rangle$, certain logical states require more single qubit gates than others to be prepared, leading to worse error rates. Once we have these circuits, we decompose them into ion-trap gates using the identities in Figure 5.2: where $RX(\theta) = exp(-i\frac{\theta}{2}X)$, $RY(\theta) = exp(-i\frac{\theta}{2}Y)$, and $XX(\theta) = exp(-i\theta XX)$. $s, v = \pm 1$ and we choose the sign to cancel as many single qubit gates as possible [60]. We think of our gates being applied through a Rabi frequency which is evolved for a time such that $\theta = \Omega_R t$. One of the benefits of the ion-trap architecture is that the two qubit Mølmer-Sørensen gates, $XX(\pi/4)$ [38, 59] can be applied between distant qubits in the chain [106, 62, 107, 108, 109]. This allows us to not only avoid the SWAP gates that other architectures rely on, but also gives us the freedom to label our ions as we desire. We will use this freedom reduce the impact of crosstalk in Section 5.3. It is on these ion trap circuits that

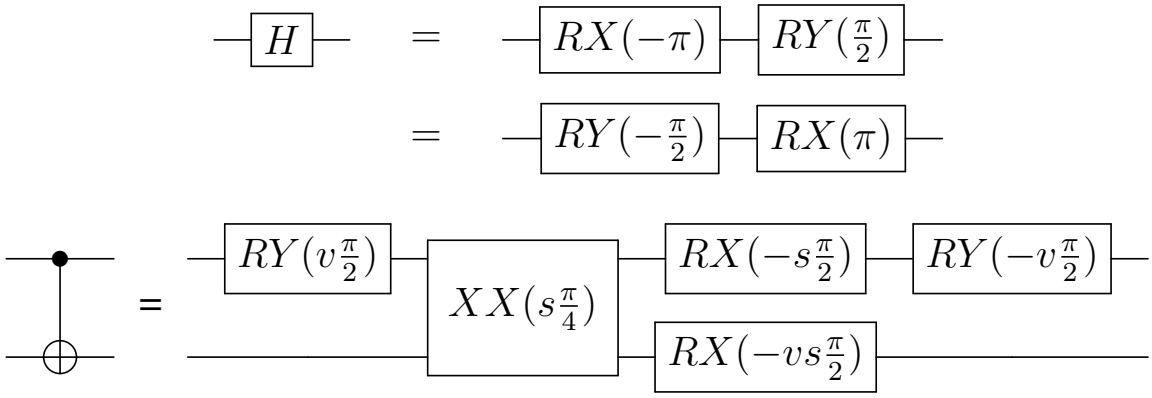


Figure 5.2: Ion trap gate compilations of CNOT and H in terms of one- and two-qubit Pauli rotations [60]. The choices of $s, v \in \{\pm 1\}$ represent degrees of freedom that only affect the global phase.

we apply our noise models.

5.2 Noise models

The most general error model we consider is the depolarizing model, which applies random Pauli errors after gates:

$$E_{1d} = \{\sqrt{1-p_{1d}}I, \sqrt{p_{1d}/3}X, \sqrt{p_{1d}/3}Y, \sqrt{p_{1d}/3}Z\}$$
$$E_{2d} = \{\sqrt{1-p_{2d}}I, \sqrt{p_{2d}/15}IX, \ldots \sqrt{p_{2d}/15}ZZ\}$$

Our model holds the single qubit error rate as one-tenth of the two qubit error rate:

$$P_{1d} = \frac{1}{10}P_{2d}.$$

This model has been studied many times and we mention it to provide a frame of reference when considering our results in relation to other work in the field.

The three ion trap specific error types we consider are T_2 dephasing, overrotation, and crosstalk. The first of these errors is an idling error, and the remaining two are gate errors. For all models we only consider stochastic channels due to limitations on memory within simulations for a logical two-qubit gate.

5.2.1 T_2 Dephasing

We only consider idling error in the form of T_2 dephasing due to the long T_1 times in trapped ion systems. In this chapter we look at T_2 times in the range of

$$0 \leq \frac{1}{T_2} \leq 2\ s^{-1}$$

for gate times of $10\mu s$ and $200\mu s$ for single and two qubit gates respectively. We also will allow for parallel single qubit gates, but only allow one two-qubit gate to be active at a time. This restriction is pessimistic given recent implementations of parallel gates [110]. During the application of all one- and two-qubit gates we model a single-qubit dephasing error on each of the idling qubits (qubits that are not affected by the gate in operation) with the Kraus channel

$$E_{idle} = \{\sqrt{1-p_i}I, \sqrt{p_i}Z\}, \tag{5.1a}$$

where

$$p_i = \frac{1}{2}\left(1 - \exp\left[-\frac{1}{2}\frac{T_{idle}}{T_2}\right]\right), \tag{5.1b}$$

and T_{idle} is the idling time of the particular qubit.

5.2.2 Gate Error

The next form of error we will consider are gate errors inspired by overrotation. These errors occur on any one or two qubit gate applied in ion traps, are one of our dominant sources of error [55], and can stem from sources such as incorrect timing or miscalibrated laser intensities that lead to fluctuations in Rabi frequency. For single qubit gates, we can use composite pulse sequences to suppress the error [111], but for two qubit gates these sequences take prohibitive amounts of time. In certain cases these multi-qubit gate errors can be dealt with effectively as we will discuss in the next section, but in general we will

have to rely on error correction to fix these errors. In the overrotation error model, the gate error following some Pauli rotation gate G has the form,

$$\varepsilon_G(\rho) = \kappa \cdot \varepsilon_G^c(\rho) + (1-\kappa) \cdot \varepsilon_G^s(\rho) \tag{5.2}$$

where ε_G^c and ε_G^s are coherent and stochastic overrotation channels with equal fidelity given by,

$$\begin{aligned} \varepsilon_G^c(\rho) &= \exp(-i\epsilon G)\rho \exp(i\epsilon G) \\ \varepsilon_G^s(\rho) &= \cos^2(\epsilon) I\rho I + \sin^2(\epsilon) G\rho G. \end{aligned} \tag{5.3}$$

In this chapter we focus on stochastic gate error channels where $\kappa = 0$. We model the stochastic overrotation channels in the Clifford simulation as

$$E_{MS} = \{\sqrt{1-p_{MS}}I, \sqrt{p_{MS}}XX\},\ p_{MS} = \sin^2(\epsilon_{MS})$$

after Mølmer-Sørensen gate, and

$$E_{1q} = \{\sqrt{1-p_{1q}}I, \sqrt{p_{1q}}P\},\ p_{1q} = \sin^2(\epsilon_{1q})$$

after single-qubit rotation gates $G \in \{R_X, R_Y, R_Z\}$. This error model is less damaging in general than the coherent case, but these errors must be mitigated through error correction instead of creative compiling.

The challenge of correcting coherent overrotation errors in ion traps is interesting because the errors are in fact invertible. If one is able to apply the correct channel to the data, the error can have its damage undone. This is in contrast with stochastic error channels which cannot be inverted and as a result require projective measurement and correction in order to be dealt with. The technique of *stabilizer slicing* handles coherent overrotations

by taking advantage of the underlying stabilizer state nature of our logical codestates in order to direct overrotations against each other [112]. In this way we can eliminate the impact of the errors stemming from stabilizer measurement before they can even be seen. Due to their symmetries being easily broken down into weight-2 operators the Shor codes are best suited for implementing stabilizer slicing using present day physical gates. The Bacon-Shor code can also implement slicing, but at slightly lower effectiveness over multiple rounds as the gauge wanders in time. Ref. [112] shows that with increasing coherence in the error stabilizer slicing yields an improvement in single logical qubit error correction circuits for Bacon-Shor-13 and constant performance with coherence for Surface-17.

5.2.3 Crosstalk

Finally, crosstalk is an issue that leads to pairwise correlated errors when applying our native entangling gate, the Mølmer-Sørensen gate. When an entangling gate is applied, a global beam is applied to the chain, and individually addressed beams are applied to the involved qubits. These addressed beams can have some degree of overlap with the neighboring qubits. For single qubit gates, this can easily be handled by narrowband or passband composite pulses [95]. For two-qubit gates, this leads to a possibility for small Mølmer-Sørensen type errors between the involved qubits and any of these nearest neighbors. We model this effect through applying a Kraus channel to all qubit pairs $\{q_i, q_n\}$ where q_i is a qubit involved in the desired Mølmer-Sørensen gate, and q_n is a qubit that neighbors either of the involved qubits in the physical ion chain. These pairs are shown in Figure 5.3. For each of these pairs the following Kraus channel is applied:

$$E_{crosstalk} = \{\sqrt{1-p_c}II, \sqrt{p_c}XX\}, \tag{5.4a}$$

where

$$p_c = \sin^2(\frac{\Omega_c}{\Omega_R} \times \frac{\pi}{4}). \tag{5.4b}$$

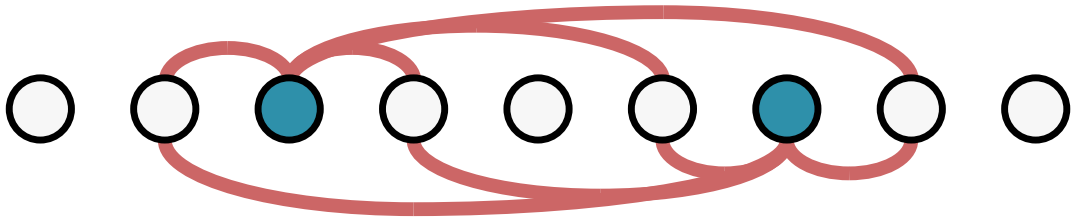


Figure 5.3: The first order crosstalk errors, shown in red, which occur during a Mølmer-Sørensen gate on the qubits shaded in blue.

Ω_c/Ω_R is the two-qubit gate crosstalk Rabi ratio, which gives the ratio of the Rabi frequency experienced by these crosstalk pairs and the Rabi frequency of the intended gate. Under this model, a single Mølmer-Sørensen gate can lead to 8 possible first order Mølmer-Sørensen type crosstalk errors when the qubits are well separated. If the intended gate is being applied on two qubits with only a single qubit separating them, the effect increases dramatically and crosstalk errors featuring this central qubit occur at four times their usual rate. These large scale correlated errors can cause issues with fault tolerance, and chain orderings which do not account for them may have possible first order crosstalk events that lead to a logical error, as shown in Figure 5.4. We explain methods for avoiding these damaging crosstalk events in the following section. In this chapter we consider stochastic crosstalk, however in the case of coherent crosstalk one can use Pauli conjugation to control the impact of these errors [113], along with dynamical decoupling methods [114].

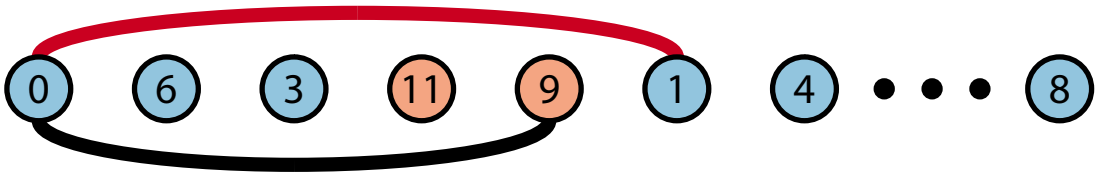


Figure 5.4: In this figure we show a possible chain to encode Bacon-Shor-13, where the data qubits are blue and the ancillae are red. When the Mølmer-Sørensen gate (denoted in black) is applied between qubits 0 and 9 as part of the $Z_0Z_3Z_1Z_4Z_2Z_5$ stabilizer, there is a first order crosstalk event which causes an XX type error (in red) between qubits 0 and 1. This error will cause a logical error as Bacon-Shor-13 cannot differentiate between it and the weight-1 X-type error on qubit 2, and when this correction is implemented we would have applied a full X-type logical error.

5.3 Fault-tolerance to Crosstalk in Ion Chains

In practice when we apply a two-qubit operation between the information stored in the ith and jth ion on an ion chain, the four neighboring ions $i-1, i+1, j-1$ and $j+1$ may be affected by the laser beam, which would introduce undesired crosstalk. In our stochastic model, a full XX error can happen between the qubits stored on the following pairs of ions $(i-1,i)$, $(i+1,i)$,$(j-1,i)$, $(j+1,i)$, $(i-1,j)$, $(i+1,j)$, $(j-1,j)$, $(j+1,j)$, which we will refer to as the crosstalk pairs. The XX error happens on each pair with probability p_c defined in Eq. 5.4b and pairs for which crosstalk can induce a logical error will be referred to as bad crosstalk pairs.

There are multiple classes of bad crosstalk errors. Crosstalk errors that impact two data qubits, as in Figure 5.4, can clearly cause a distance drop. Additionally, crosstalk errors which apply to both a data qubit and an ancillary qubit such that the X error propagates back to a data qubit can also cause issues. The last case is one in which two ancillary qubits have a crosstalk error between them which causes X errors to propagate to the data, causing a logical error. This type of error is avoided in our circuit by having all of our stabilizer measurements serialized, so the correlated errors do not propagate in dangerous ways. Our particular circuit compilation is also set up so that crosstalk errors are never conjugated into weight-2 Z-type errors, so Z-type logical errors are not as much of a concern in our crosstalk pairs.

In order to be robust against these crosstalk errors when implementing a small quantum error-correcting code on an ion trap quantum computer, we try to find an optimal mapping of qubits on the linear ion chain such that a single XX error event on any of the possible crosstalk pairs does not directly lead to a failure on the encoded logical state. Our problem can be formulated using the graph theory language: we construct a graph $G = (V, E)$, where the vertex set V is the set of qubits, and the edge set E consists of pairs of qubits such that mapping these two qubits as neighbors will not lead to any bad crosstalk pairs.

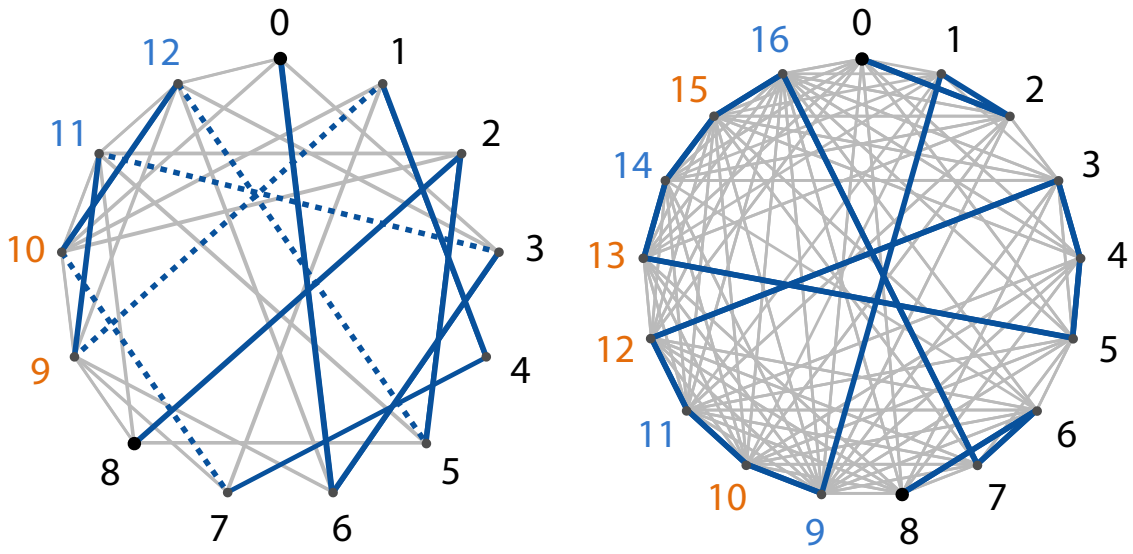


Figure 5.5: Graphs for Bacon-Shor-13 (left) and Surface-17 (right) where the edges correspond to qubits which could be neighbors without leading to distance-damaging crosstalk errors. We use dynamic programming to find a Hamiltonian path for each graph (blue lines) which also minimizes operation time. A Hamiltonian path is not possible for Bacon-Shor-13, so we must add in extra connections (dotted lines) which introduce distance-damaging crosstalk errors. Data qubits are labeled in black and ancillary qubits are labeled blue(orange) to indicate they measure $X(Z)$- type stabilizers.

One can find an ion chain without any bad crosstalk pairs by first finding a path

$$q_{i_1} \rightarrow q_{i_2} \rightarrow \cdots q_{i_j} \rightarrow \cdots \rightarrow q_{i_{|V|}}$$

in the graph that covers each vertex (qubit) exactly once, then mapping the qubit q_{i_j} to the jth ion of the chain. In Figure 5.5 we show the graphs and corresponding path solutions for Bacon-Shor-13 and Surface-17. Note that the ancillary qubits have more connections, as the way we compile our circuits means that data-data crosstalk is the most damaging effect.

The problem of traversing a graph and crossing each vertex once is known as the Hamiltonian Path problem [115]. Although Hamiltonian Path is NP-complete, we can use techniques such as dynamic programming to accelerate brute-force searching [116]. See Section 5.5 for further details.

5.3.1 Best Chains for Different Codes

Since the structure of stabilizer measurements is code-specific, the constraints that the proper ion chain needs to satisfy are also different. Consequently each chain ends up being different. Also note that all the chains presented in this section are also designed to minimize execution time for the corresponding circuit as a secondary constraint. Data qubits are depicted in black and ancillary qubits are labeled blue(orange) to indicate they measure $X(Z)$- type stabilizers.

- **Surface-17**

  0 2 1 9 10 11 12 3 4 5 13 14 15 16 7 6 8

- **Bacon-Shor-13**

  0 6 3 11 9 1 4 7 10 12 5 2 8

- **Shor-6X2Z**

  0 2 1 11 12 9 13 3 4 5 14 10 15 6 7 8 16

- **Shor-6Z2X**

  3 11 0 6 12 1 7 13 9 10 14 4 15 2 8 16 5

For Bacon-Shor-13 and the Shor codes unfortunately there do not exist any ion chain arrangement that could avoid all logical crosstalk errors, so the above chains are the ones which minimizes the impact of crosstalk errors to the system.

These codes do not have valid distance-preserving chains because they feature large weight-6 stabilizers. This means the two qubits neighboring the ancilla for these stabilizers must be acceptable crosstalk pairs with a large number of other qubits, and within the solution space provided by these small codes, there simply is not enough freedom to find a valid ordering. An alternative approach is to add additional spacer ions that could also be used for sympathetic cooling [117, 118, 119, 120]

5.4 Results and Discussion

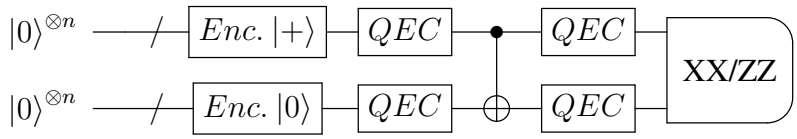


Figure 5.6: The circuit that we simulate for each code. From the $XX(ZZ)$ measurements we can gauge the code's performance in generating our desired state of $\Phi_L^+ = \frac{1}{2}(|00\rangle_L + |11\rangle_L)$.

In order to assess the performance of a code against a given error model, we use a stabilizer method with importance sampled error [22] to simulate a circuit featuring both an X and Z basis state preparation, along with the ex-Rec of a transversal CNOT, as shown in Figure 5.6. By measuring the XX and ZZ parities of the output state, we can assess the performance of the code in a way which is experimentally implementable while also including only fault tolerant circuits. Logical Y cannot be measured fault tolerantly since we are unable to do a round of classical correction after measuring the data qubits in the Y basis. One thing to note is that the Shor's code variants do not possess a fault-tolerant H. As a result, in Bacon-Shor-13 and Surface-17 both bases are similarly difficult to prepare, while in the Shor's code variants one basis has an encoding circuit and the other requires projective preparation. Despite this lack of a fault-tolerant H gate, since each Shor's code variant is CSS we can measure them in both the X and Z bases. We use these measurements to define a circuit level error version of pseudothreshold where the logical performance on this circuit is compared to the error rate for the unencoded CNOT:

$$p_{phys} = p_{2q} + 4 \cdot p_{1q} + 8 \cdot p_{crosstalk} + 2 \cdot p_{idle} \tag{5.5}$$

where

$$p_{idle} = \frac{1}{2}\left(1 - \exp\left[-\frac{t_{2q} + 3t_{1q}}{2T_2}\right]\right)$$
$$p_{crosstalk} = \sin^2\left(\frac{\Omega_c}{\Omega_R} \times \frac{\pi}{4}\right) \tag{5.6}$$

and $p_{2q} = 10\ p_{1q}$, T_2, and Ω_c/Ω_R are the three error parameters we will be studying. We include a factor of two on p_{idle} because there are at least two qubits experiencing the idling, and the factor of eight on the crosstalk probability comes from the idea that there are eight possible first order crosstalk pairs, as shown in Figure 5.3. Due to this non-standard definition of pseudothreshold, certain pseudothresholds will seem very low because the comparison we are making includes the impact of preparation circuits and transversal gates.

5.4.1 Depolarizing Error Model

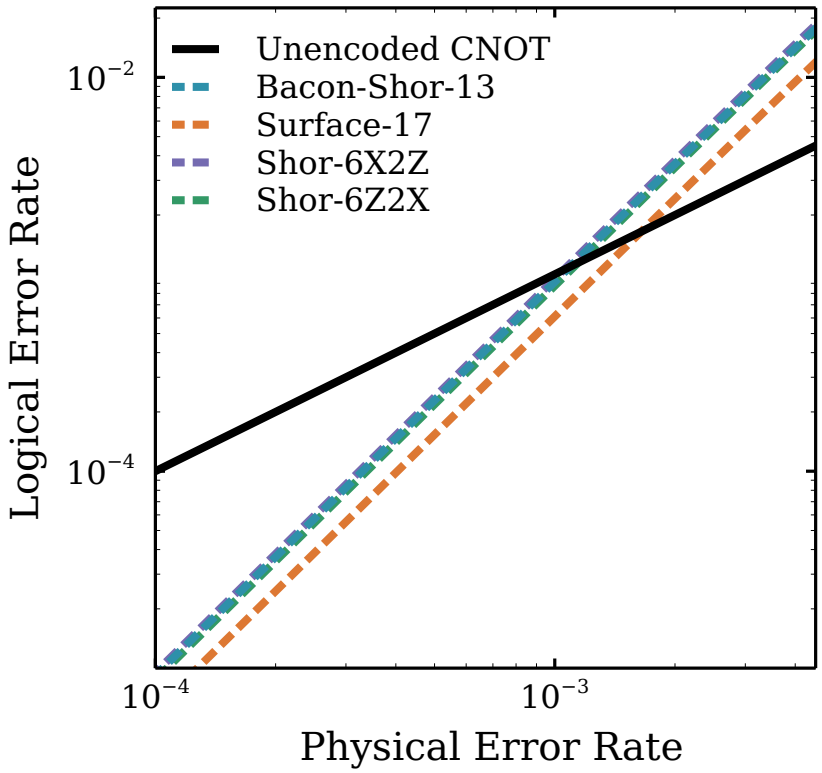


Figure 5.7: In this plot we compare the physical two qubit error rate to the error rate on the circuit in Figure 5.6 under the standard depolarizing model described in Section 5.2.

In Figure 5.7 we consider the gate depolarizing error model defined in the beginning of Section 5.2, with depolarizing errors directly following all gates but not acting on idles. Surface-17 outperforms all other codes due to its ability to correct both types of error. Bacon-Shor-13 does not have the same ability to correct two qubit errors, and both Shor's code variants share this weakness in one direction. These codes are more effective than Bacon-Shor-13 at correcting one side of error, however their projective preparation requirement for one of the logical bases cancels out this benefit. Shor-6Z2X outperforms its counterpart because the 6X2Z form of the circuit uses 64 additional single qubit gates, leading to extra error locations.

5.4.2 Ion Trap Error Models

We now consider the remaining error models mentioned in Section 5.2. In Figure 5.8 we present a series of phase diagrams indicating the transition between regions in which different codes are optimal choices. For the Mølmer-Sørensen vs. idling plots (a),(d) there are no crosstalk errors, and in the Mølmer-Sørensen vs. Rabi ratio plots (b),(e) we have set $T_2 = \infty$. However in the idling vs. Rabi ratio plots (c),(f) we have set $p_{2q} = 10\, p_{1q} = 0.0001$ since we believe it is unrealistic for there to be no error on the qubits involved in a gate.

Code Performance

In Figure 5.8 we present plots for a variety of error models showing optimal codes and circuit-level pseudothreshholds. There are a few key features that distinguish the codes.

First, Surface-17 performs very poorly under gate errors, and only begins to outperform Bacon-Shor-13 and the two Shor's code variants when the other two error models are strong relative to overrotation. The surface code's advantage on the other two error models is more pronounced in the case of crosstalk. This is due to the surface code allowing for a fault-tolerant chain ordering. For the T_2 times we consider it is only relevant when overrotation

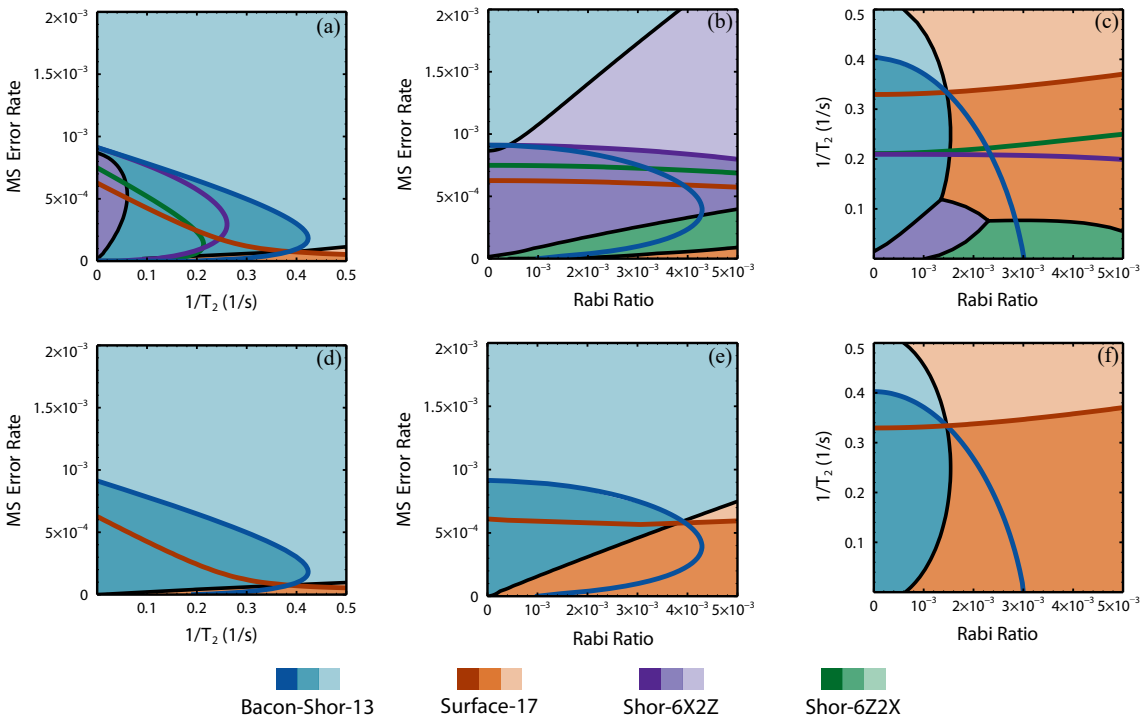


Figure 5.8: Figures showing best performing codes and pseudothresholds for different error models and sets of codes. In (a,b,c) we are comparing codes in the set {Surface-17, Bacon-Shor-13, Shor-6X2Z, Shor-6Z2X}, while in (d,e,f) we restrict our set to only consider codes with a transversal Hadamard gate, {Surface-17, Bacon-Shor-13}. In (a,d) we look at the intersection of overrotation error (parameterized by the two qubit gate error) and T_2 dephasing, in (b,e) we look at overrotation and crosstalk, and in (c,f) we look at T_2 dephasing and crosstalk with a background overrotation characterized by a Mølmer-Sørensen error rate of 10^{-4}. The colored regions indicate which code is optimal at those error parameters, with darker shading implying the code is outperforming a physical CNOT. The colored curves are the pseudothreshold curves for which the logical error rate is equal to the physical error rate in Eq. 5.5 and the black curves are borders between regions in which different codes are preferred.

errors are practically nonexistent. As would be expected from this trend, the plot in which it performs best is the idling versus crosstalk plot, in which overrotation error is minimal.

Second, crosstalk seems to be the only situation in which Bacon-Shor-13 does not excel in our comparison. Some of this is from the fact that Bacon-Shor-13 has a fault-tolerant preparation circuit for both bases, while both the Shor's code variants need projective preparation for one basis and Surface-17 needs it for both. Bacon-Shor-13 is also good at handling overrotations as they become more coherent, so for our ion-trap error models

we believe it to be the best choice as long as we can control crosstalk via other means.

Finally, Figure 5.8b the data shows that Shor-6X2Z is better at dealing with crosstalk than Shor-6Z2X. This seems unintuitive as Shor-6Z2X is optimized for catching the X-type errors that crosstalk is slightly biased towards, however due to its stabilizer structure, Shor-6X2Z is able to completely ignore a number of weight-2 X errors on data qubits due to them being in its stabilizer group. This effect leads to Shor-6X2Z being better at correcting pairwise correlated X-type errors even though Shor-6Z2X is preferable for single data qubit errors.

Other than best performing codes, these plots also include pseudothresholds. Due to our particular metric based on Figure 5.6, they look different than would be expected based on other work in this area. This discrepancy is especially noticeable in the case of T_2 dephasing, where it can be seen that as other error sources approach zero, the pseudothreshold decreases significantly. Most definitions of pseudothreshold with respect to T_2 dephasing compare the dephasing timescale of the physical qubit to that of the encoded qubit, whereas our definition compares the two in a situation where the encoded circuit is expected to operate for significantly longer in order to implement the same logical operation cleanly. This depresses the crossover point to be below error rates for which the error corrected qubit lasts longer than the physical one.

Selected Logical Error Biases

Due to the format of our data, we are able to separately consider the rates at which the XX and ZZ parity is violated, allowing us to consider the bias of error at the logical level due to these asymmetrically structured error models. In this section we will highlight Bacon-Shor-13 and Surface-17, the logical bias plots for both sides of Shor's code can be found in Section 5.6. The value we show in this plot is

$$Bias_{ZZ} = \frac{\langle 1 - Z_{L1}Z_{L2} \rangle}{\langle 1 - X_{L1}X_{L2} \rangle + \langle 1 - Z_{L1}Z_{L2} \rangle} \tag{5.7}$$

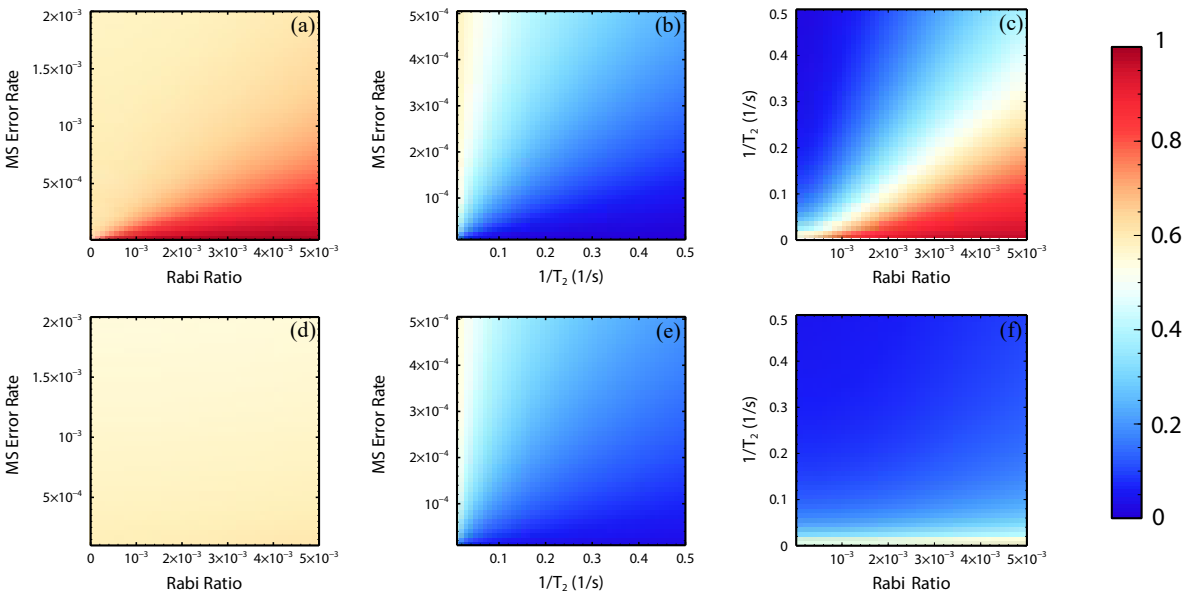


Figure 5.9: Plots of $Bias_{ZZ}$, which is defined in Equation 5.7 We present data for Bacon-Shor-13 (top row) and Surface-17 (bottom row) over a series of different error models. In the rightmost plots there is also a background overrotation error rate of $p_{2q} = 10\, p_{1q} = 0.0001$.

where $\langle 1 - Z_{L1}Z_{L2} \rangle$ and $\langle 1 - X_{L1}X_{L2} \rangle$ are the rates at which the ZZ and XX parities are violated.

From the plots in Figure 5.9 we can see that the most strongly biased error is T_2 dephasing. While both dephasing and crosstalk errors always appear as a specific type, crosstalk errors during Z-type stabilizers are often found sandwiched by single qubit gates which convert them to Z-type errors. This factor, along with our crosstalk mitigation techniques preventing dangerous X-type logical error pairs from being adjacent in our chains, leads to crosstalk being only slightly biased towards violating ZZ in the cases of Shor-6Z2X and Surface-17. For Bacon-Shor-13 there was not enough freedom for us to implement any of our crosstalk techniques, so the native X-type bias is quite strong. In Shor-6X2Z, the larger number of X stabilizers means that there is an asymmetry in the number of error locations in which a crosstalk error would be conjugated into a Z-type error, leading to a significant X-type bias as well. From these results we can see that the way in which we try to prevent crosstalk through chain ordering has a strong impact on the bias at the logical

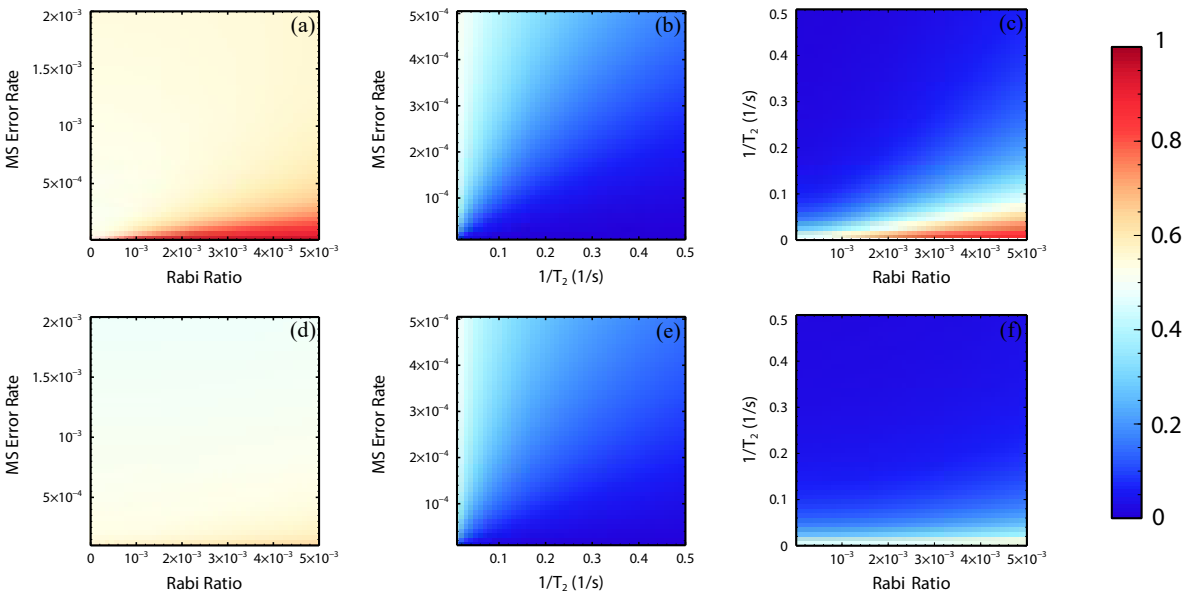


Figure 5.10: Plots of $Bias_{ZZ}$, which is defined in Equation 5.7 We present data for Shor-6X2Z (top row) and Shor-6Z2X (bottom row) over a series of different error models. In the rightmost plots there is also a background overrotation error rate of $p_{2q} = 10\,p_{1q} = 0.0001$.

level. It is possible that by intelligently picking this bias to interface with the underlying error models, we would be able to create an effective error model at the logical level with significant structure.

5.5 Dynamic Programming

A standard approach for finding the Hamiltonian path of a graph $G = (V, E)$ is dynamic programming [116]. In this method, one determines whether there exists, for each subset $S \subset V$ of vertices and each vertex $v \in S$, a Hamiltonian path that covers S and ends at v. For each (S, v), a path exists if and only if a path exists for $(S - \{v\}, w)$ for some $w \in S - \{v\}$ such that $(v, w) \in E$. Note that one can look up already-computed answers to avoid redundant computation. Since there are only $O(n2^n)$ number of choices of (S, v), and enumerating $w \in S - \{v\}$ takes $O(|V|)$ time, the total time complexity is $O(|V|^2 2^{|V|})$.

For our problem, it is possible that an ion chain without bad crosstalk does not exist. In this case the problem becomes finding the path that touches each vertex once and requires

the fewest additional edges added to the graph. To use dynamic programming, one can ask the following question instead: for each subset $S \subset V$ of vertices, $v \in S$, and non-negative integer n, does there exist a path that covers S while touching each vertex once, ends at v, and only requires n extra edges added to the graph. For each tuple (S, v, n), a solution exists if and only if one of the following two cases happens:

1. Solutions exist for $(S - \{v\}, w, n - 1)$ for some $w \in S - \{v\}$, however $(v, w) \notin E$. As a result an edge must be added which introduces a bad crosstalk pair.

2. Solutions exist for $(S - \{v\}, w, n)$ for some $w \in S - \{v\}$ such that $(v, w) \in E$, then the edge that is added does not introduce any bad crosstalk pairs.

n has a trivial upper bound $|V|$ since we can definitely use $|V|$ paths to cover the vertex set. Therefore the time complexity is $O(|V|^3 2^{|V|})$.

5.6 Shor's Codes Logical Bias Plots

In Figure 5.10 we see the same data as shown in Figure 5.9 but for Shor-6X2Z and Shor-6Z2X. The only major difference between these two codes is that Shor-6X2Z has crosstalk that preserves its bias towards X, while Shor-6Z2X is more neutral. They both also have a slightly more Z-biased error rate in the overrotation vs. crosstalk plots. As mentioned in the main text, the discrepancy in bias for crosstalk can be explained by the amount of time in the circuit during which a crosstalk error would end up being conjugated into a Z-type error, along with the lower number of X-type logical operators present for Shor-6Z2X.

5.7 Conclusion

In this chapter we have shown that there are a wide variety of optimal codes when considering different error sources, indicating the importance of being able to accurately benchmark a system and find the error models and parameters which describe it. In the depolarizing error model Surface-17 clearly outperforms all other codes, however when considering

physically realistic error models Bacon-Shor-13 and Shor's code variants perform better. Due to the all to all connectivity present in ion trap systems, the surface code is not benefited significantly from its locality. Interestingly, even when considering superconducting systems that have nearest neighbor interactions, other codes continue to outperform the surface code when considering experimental constraints [121].

We also provide evidence for how damaging crosstalk errors really are, further justifying the efforts in looking for methods outside of QECCs for solving it. Implementing a physical or pulse level solution to mitigate the effects of crosstalk will be vital in allowing us to consider a wider variety of codes on our systems. If crosstalk can be lessened or reduced coherently, Bacon-Shor-13 seems to be well suited to solving the other errors present in ion trap systems. We also see that by making choices about how our chains are ordered, we can affect the logical error biases, which could be used to make a more optimized asymmetric code in the future.

CHAPTER 6

A NUMERICAL STUDY OF BRAVYI-BACON-SHOR AND SUBSYSTEM HYPERGRAPH PRODUCT CODES

This chapter contains results from

Muyuan Li, Theodore J. Yoder, arXiv preprint arXiv: 2002.06257 (2020)

Two-dimensional topological error-correcting codes are extremely attractive models of quantum error-correction. Structurally, low-weight stabilizers – just weight four for the surface code and weight six for the most popular color code – that are also local in the plane make for simple fault-tolerant syndrome measurement circuits. In turn, this simplicity leads to surprisingly high thresholds [122] compared to, say, concatenated codes [19].

On the other hand, error-correction in two dimensions is inherently limited by the Bravyi-Poulin-Terhal bound [123], which states that a two-dimensional code using N qubits to encode K qubits with code distance D must satisfy $cKD^2 \leq N$ for some universal constant c. In particular, two-dimensional codes with constant rate $K \propto N$ must have constant distance, which precludes error-correction with constant space overhead [124] in two dimensions.

These constraints on two-dimensional codes explains the recent surge of interest in quantum hypergraph product codes [125, 126], which break the plane (i.e. are *not* local in two dimensions) but in doing so achieve $K \propto N$ and $D \propto \sqrt{N}$. Given the small-set flip decoder [126], which is single-shot with an asymptotic threshold, hypergraph product codes promise quantum error-correction with constant overhead [127].

However, hypergraph product codes also have a couple of undesirable properties from a practical standpoint. First, the small-set flip decoder, although theoretically satisfactory, is likely not practical due to low thresholds even when measurements are perfect [128]. This is somewhat to be expected by analogy with classical expander codes, where the

classical flip decoder [129] is greatly outperformed by heuristic decoders, such as belief propagation [130]. It is also unclear that the small-set flip decoder works well at all on small examples suitable for near-term implementation. Second, the stabilizer weights of hypergraph product codes are relatively large, e.g. the best performing codes in [128] have stabilizers with weight 11, which necessitates a corresponding increase in fault-tolerant circuit complexity and a decrease in thresholds with respect to circuit-level noise.

Here we take an empirical look at two families of subsystem codes that, while related to hypergraph product codes, may have some advantages for near-term implementation. Because these are subsystem codes, the operators measured for error-correction are quite small – in the cases we explore here they never exceed weight six. We also demonstrate how the powerful technique of belief propagation can be applied to decode these codes.

The first family consists of the Bravyi-Bacon-Shor (BBS) codes [131]. BBS codes achieve $K, D \propto \sqrt{N}$ with just two-body measurements and are easily modified so that these measurements are local in two dimensions. Furthermore, they can be gauge-fixed to hypergraph product codes [132]. The second family consists of the "generalized Shor" codes of Bacon and Cassacino [133]. We rename these codes subsystem hypergraph product (SHP) codes, because we prove that any hypergraph product code is two SHP codes with their gauge qubits entangled. SHP codes can achieve $K \propto N$ and $D \propto \sqrt{N}$ just like hypergraph product codes. Compared to BBS codes, they have higher weight gauge operators, weight six in our instances.

We perform numerical experiments with these code families in two regimes of operation. In the small-code regime, we construct small, distance-3 codes in each class, develop fault-tolerant circuits for measuring their stabilizers, and calculate pseudothresholds for circuit noise. We find pseudotresholds of 2×10^{-3} for a $[\![21, 4, 3]\!]$ BBS code and 8×10^{-4} for a $[\![49, 16, 3]\!]$ SHP code. These results suggest that the BBS code in particular is quite a good candidate for protecting four logical qubits with a small quantum computer.

In the large-code regime, we create BBS and SHP codes from regular classical ex-

pander codes. We modify belief propagation to include measurement errors and apply it to decode these codes under an error model including data and measurement noise (but without circuit-level noise). Despite no asymptotic thresholds, compared to a surface code with similar code rate, BBS and SHP code do achieve better logical error rates per logical qubit provided sufficiently low physical error rates: $p < 10^{-6}$ for BBS codes and $p < 4 \times 10^{-4}$ for SHP codes.

6.1 Review of Bravyi-Bacon-Shor Codes

In this section, we review the Bravyi-Bacon-Shor (BBS) codes that were introduced by Bravyi [123] and explicitly constructed in [132].

Let $\mathcal{F}_2$ denote the finite field with two elements 0,1. A Bravyi-Bacon-Shor code is defined by a binary matrix $A \in \mathcal{F}_2^{n_1 \times n_2}$, where qubits live on sites (i, j) of the matrix A for which $A_{i,j} = 1$. As shown in [123, 132], given A we can define two classical codes corresponding to its column-space and row-space:

$$\mathcal{C}_1 = \text{col}(A), \tag{6.1}$$

$$\mathcal{C}_2 = \text{row}(A), \tag{6.2}$$

where $\mathcal{C}_1$ and $\mathcal{C}_2$ has code parameters $[n_1, k, d_1]$, $[n_2, k, d_2]$, generating matrices G_1 and G_2, and parity check matrices H_1 and H_2.

The notation for Pauli operators on the qubit lattice is defined as follows. A Pauli X- or Z-type operator acting on the qubit at site (i, j) in the lattice is written as $X_{i,j}$ or $Z_{i,j}$. A Pauli operator acting on multiple qubits is specified by its support S:

$$X(S) = \prod_{ij} (X_{i,j})^{S_{ij}}, \ S \in \mathcal{F}_2^{n_1 \times n_2}, \tag{6.3}$$

where $S_{ij} = 1$ implies that $A_{ij} = 1$, since qubits only exist where $A_{ij} = 1$. Similar

notations will be used throughout the rest of this chapter. We let $|A| = \sum_{ij} A_{ij}$ and $|v| = \sum_i v_i$ denote the Hamming weights of matrices and vectors.

Definition 1. [123] The Bravyi-Bacon-Shor code constructed from $A \in \mathcal{F}_2^{n_1 \times n_2}$, denoted BBS$(A)$, is an $[\![N, K, D]\!]$ quantum subsystem code with gauge group generated by 2-qubit operators and

$$\begin{aligned} N &= |A|, \\ K &= \text{rank}(A), \\ D &= \min\{|\vec{y}| > 0 : \vec{y} \in \text{row}(A) \cup \text{col}(A)\}, \end{aligned}$$

As CSS quantum subsystem codes, the gauge group of BBS codes is generated by XX interactions between any two qubits sharing a column in A and ZZ between any two qubits sharing a row in A. The gauge group can be more formally written as

$$\mathcal{G}_X^{(\text{bbs})} = \{X(S) : G_R S = 0, S \subseteq A\}, \tag{6.4}$$

$$\mathcal{G}_Z^{(\text{bbs})} = \{Z(S) : S G_R^T = 0, S \subseteq A\}, \tag{6.5}$$

where $G_R = (1, 1, \ldots, 1)$ is the generating matrix of the classical repetition code, and the subset notation $S \subseteq A$ means that S is a matrix such that, for all i, j, $S_{ij} = 1$ implies $A_{ij} = 1$.

For bare logical operators of the BBS code to commute with all of its gauge operators, each bare logical X-type operator must be supported on entire rows of the matrix and each bare logical Z-type operator must be supported on entire columns of the matrix. To express this similarly to the gauge operators above, define the parity check matrix of the classical

repetition code H_R. Then we have the sets of X- and Z-type logical operators:

$$\mathcal{L}_X^{(\text{bbs})} = \{X(S \cap A) : SH_R^T = 0\}, \tag{6.6}$$

$$\mathcal{L}_Z^{(\text{bbs})} = \{Z(S \cap A) : H_R S = 0\}. \tag{6.7}$$

Consequently, the group of stabilizers for the BBS code is the intersection of the group of bare logical operators with the gauge group:

$$\mathcal{S}_X^{(\text{bbs})} = \mathcal{L}_X^{(\text{bbs})} \cap \mathcal{G}_X^{(\text{bbs})} \tag{6.8}$$

$$= \{X(S \cap A) : SH_R^T = 0, G_1 S = 0\}, \tag{6.9}$$

$$\mathcal{S}_Z^{(\text{bbs})} = \mathcal{L}_Z^{(\text{bbs})} \cap \mathcal{G}_Z^{(\text{bbs})} \tag{6.10}$$

$$= \{Z(S \cap A) : H_R S = 0, SG_2^T = 0\}. \tag{6.11}$$

6.1.1 Constructing BBS Codes with Classical Linear Codes

In [132], the following method of constructing a BBS code from classical codes was given.

Theorem 2. *Given two classical linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ with parameters $[n_1, k, d_1]$ and $[n_2, k, d_2]$, and generating matrices $G_1 \in \mathcal{F}_2^{k \times n_1}$ and $G_2 \in \mathcal{F}_2^{k \times n_2}$, we can construct the code $BBS(A)$ by*

$$A = G_1^T Q G_2 \in \mathcal{F}_2^{n_1 \times n_2}, \tag{6.12}$$

where $Q \in \mathcal{F}_2^{k \times k}$ can be any full rank $k \times k$ matrix. Then $BBS(A)$ is an $[\![N, K, D]\!]$ quantum subsystem code with

$$min(n_1 d_2, d_1 n_2) \leq N \leq n_1 n_2, \tag{6.13}$$

$$K = k, \tag{6.14}$$

$$D = min(d_1, d_2). \tag{6.15}$$

Table 6.1: Stabilizers and a set of canonical logical operators for the $[\![21, 4, 3]\!]$ Bravyi-Bacon-Shor code constructed using the $[7, 4, 3]$ Hamming code.

Qubits	X_L	Z_L	Stabilizers
1	$X_0X_1X_2$	$Z_0Z_12Z_{17}$	$X_0X_1X_2X_3X_4X_5X_9X_{10}X_{11}X_{12}X_{13}X_{14}$
2	$X_3X_4X_5$	$Z_3Z_9Z_{16}$	$X_0X_1X_2X_6X_7X_8X_9X_{10}X_{11}X_{15}X_{16}X_{17}$
3	$X_6X_7X_8$	$Z_6Z_{15}Z_{18}$	$X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}X_{18}X_{19}X_{20}$
4	$X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}$	$Z_3Z_4Z_9Z_{13}Z_{16}Z_{19}$	$Z_6Z_{15}Z_{18}Z_3Z_9Z_{16}Z_4Z_{13}Z_{19}Z_7Z_{10}Z_{14}$
			$Z_6Z_{15}Z_{18}Z_0Z_{12}Z_{17}Z_4Z_{13}Z_{19}Z_1Z_5Z_8$
			$Z_3Z_9Z_{16}Z_0Z_{12}Z_{17}Z_4Z_{13}Z_{19}Z_2Z_{11}Z_{20}$

The matrix $Q \in \mathcal{F}_2^{k\times k}$ represents the non-uniqueness of the generating matrices, and adjusting Q would only affect the number of physical qubits in BBS(A). It is easy to see that col(A) = row(G_1) = $\mathcal{C}_1$, and row(A) = row(G_2) = $\mathcal{C}_2$, and the conclusions in the theorem about the code parameters follow.

6.1.2 Example: A $[\![21, 4, 3]\!]$ Bravyi-Bacon-Shor Code

The $[7, 4, 3]$ Hamming code is generated by

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \; H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Using $Q = \begin{pmatrix} 0&0&1&0 \\ 0&1&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \end{pmatrix}$ we can construct a $[\![21, 4, 3]\!]$ Bravyi-Bacon-Shor code $A = G^T Q G$:

$$A = \begin{bmatrix} 0&0&1&0&0&1&1 \\ 0&1&0&1&0&1&0 \\ 1&0&0&0&1&1&0 \\ 0&1&0&0&1&0&1 \\ 0&0&1&1&1&0&0 \\ 1&1&1&0&0&0&0 \\ 1&0&0&1&0&0&1 \end{bmatrix},$$

which minimizes the number of qubits used. We construct a canonical set of bare logical operators for the four logical qubits encoded in the $[\![21, 4, 3]\!]$ code along with a set of stabilizers generators, as shown in TABLE 6.1. Note that while qubit 4 has high weight bare logical operators due to the construction that we have chosen, it can still suffer from weight three logical operators, such as $Z_4 Z_{13} Z_{19}$, and so its error rate has the same slope as the others.

We estimated the performance of this code by simulating it under circuit level standard depolarizing noise, where Pauli channels with Kraus operators

$$\begin{aligned} E_{1q} &= \{\sqrt{1-p}I, \sqrt{\frac{p}{3}}X, \sqrt{\frac{p}{3}}Y, \sqrt{\frac{p}{3}}Z\}, \\ E_{2q} &= \{\sqrt{1-p}I, \sqrt{\frac{p}{15}}IX, \ldots \sqrt{\frac{p}{15}}ZZ\}, \end{aligned} \tag{6.16}$$

are applied after each 1- and 2-qubit gate in the circuit, respectively. We call $p \in [0, 1]$ the physical error rate. Assuming the code is fault-tolerantly prepared into its logical $|0000\rangle$ state, we simulated the circuit of error correction and destructive measurement of data qubits with single qubit memory errors added before error correction. The same error rate is used across the circuit for memory errors, gate errors, and measurement errors.

The results are shown in FIG. 6.1. Note that since the BBS codes can be considered a compass code in 2-dimensions [63], to create a fault-tolerant circuit for syndrome extraction it suffices to use a single ancillary qubit for each of the weight-12 stabilizers as listed in TABLE 6.1. Hence the total number of qubits required to perform fault-tolerant syndrome extraction for this code is $21 + 6 = 27$. We perform the syndrome extraction once and if the syndrome is trivial, we stop and no correction is needed. If the syndrome is not trivial, we measure the syndrome again and decode with the outcome.

From FIG. 6.1 we can see that qubit 4 performs slightly worse than qubits 1-3, due to the fact that its higher weight logical operators have more chance of anti-commuting with dressed logical operators than qubits 1-3.

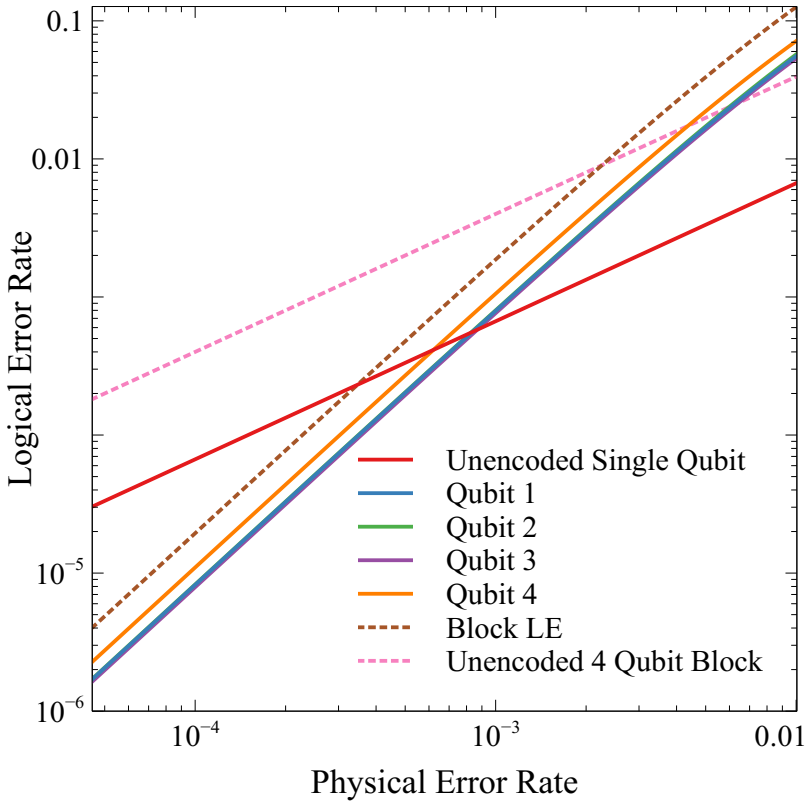


Figure 6.1: Simulated performance of the $[\![21, 4, 3]\!]$ BBS code under circuit level depolarizing error, with one ancillary qubit per stabilizer for fault-tolerant syndrome extraction. The block pseudothreshold for the code block with 4 encoded logical qubits is 2.3×10^{-3}, while the per logical qubit pseudothreshold for qubits 1-3 is 8.7×10^{-4}.

6.2 Another Family of Subsystem Hypergraph Product Codes

In this section we take a look at the "generalized Shor" codes in Bacon and Casaccino [133] from a new perspective. In particular, we find that these codes are in a sense the most natural subsystem hypergraph product codes because two of them, without ancillas, can be gauge-fixed to a hypergraph product code and, conversely, any hypergraph product code can be gauge-fixed into two generalized Shor codes. We therefore refer to generalized Shor codes as subsystem hypergraph product (SHP) codes.

Contrast SHP codes with BBS codes, which can also be gauge-fixed to hypergraph product codes [132]. Gauge-fixing BBS codes requires ancillas and the result is only a certain subset of all hypergraph product codes with less than constant rate.

6.2.1 Hypergraph Product Codes

To facilitate our proofs, we review the hypergraph product code construction briefly in this section.

Definition 3. [125] Let $H_1 \in \{0,1\}^{n_1^T \times n_1}$ and $H_2 \in \{0,1\}^{n_2^T \times n_2}$. The hypergraph product (HGP) of H_1 and H_2 is a quantum code HGP(H_1, H_2) with stabilizers

$$S_X^{(\text{hgp})} = \left(H_1 \otimes I_{n_2}, I_{n_1^T} \otimes H_2^T \right), \tag{6.17}$$

$$S_Z^{(\text{hgp})} = \left(I_{n_1} \otimes H_2, H_1^T \otimes I_{n_2^T} \right). \tag{6.18}$$

By Eq. (6.17) we mean that each vector $v \in \mathcal{F}_2^N$ in the rowspace of the matrix on the righthand side indicates an X-type Pauli operator $X^v := \prod_{i=1}^N X_i^{v_i}$ in the stabilizer group. Likewise with Z-type operators in Eq. (6.18). Similar notation will be used throughout this section.

Treating H_1 and H_2 as parity check matrices, we have two classical codes $\mathcal{C}_1 = \ker(H_1)$ and $\mathcal{C}_2 = \ker(H_2)$ with some parameters $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$, respectively. Likewise,

treat H_1^T and H_2^T as parity check matrices of the "transpose" codes $\mathcal{C}_1^T = \ker(H_1^T)$ and $\mathcal{C}_2^T = \ker(H_2^T)$ with respective parameters $[n_1^T, k_1^T, d_1^T]$ and $[n_2^T, k_2^T, d_2^T]$. Because of the rank-nullity theorem

$$n_i - k_i = n_i^T - k_i^T \tag{6.19}$$

for $i = 1, 2$. The hypergraph product code HGP(H_1, H_2) then has parameters [125]

$$[\![n_1 n_2 + n_1^T n_2^T, k_1 k_2 + k_1^T k_2^T, D]\!], \tag{6.20}$$

where

$$D = \begin{cases} \min(d_1, d_2), & k_1^T = 0 \text{ or } k_2^T = 0 \\ \min(d_1, d_2, d_1^T, d_2^T), & \text{otherwise} \end{cases}. \tag{6.21}$$

Moving on, we notice that there are $n_1 n_2 + n_1^T n_2^T$ qubits in HGP(H_1, H_2) that we lay out on two square lattices, an $n_1 \times n_2$ lattice referred to as the "large" lattice, denoted L, and an $n_1^T \times n_2^T$ lattice referred to as the "small" lattice, denoted l. See Fig. 6.2. Despite the names, the small lattice need not contain fewer qubits than the large lattice, although typically (e.g. in random constructions of classical LDPC codes [20]) $n_i^T \approx n_i - k_i < n_i$ and this is the case.

We label qubits in these lattices in row major fashion. Thus, a (row) vector $r^T \otimes c^T$ for $r \in \{0,1\}^{n_1}$ and $c \in \{0,1\}^{n_2}$ indicates exactly the qubits that are both in the rows indicated by r and in the columns indicated by c of the large lattice. Qubits in the large lattice are labeled first, i.e. $1, 2, \ldots, n_1 n_2$, followed by qubits in the small lattice, i.e. $n_1 n_2 + 1, \ldots, n_1 n_2 + n_1^T n_2^T$.

For later purposes, we point out some subgroups of the stabilizer group. For instance, certain stabilizers of HGP(H_1, H_2) are supported entirely on the large lattice. Because the rowspace of $S_X^{(\text{hgp})}$ represents all X-type stabilizers, if $x \in \{0,1\}^{n_1^T}$, $c \in \ker(H_2) = \mathcal{C}_2 \subseteq \{0,1\}^{n_2}$, then

$$(x^T \otimes c^T) S_X^{(\text{hgp})} = \left(x^T H_1 \otimes c^T, 0\right) \tag{6.22}$$

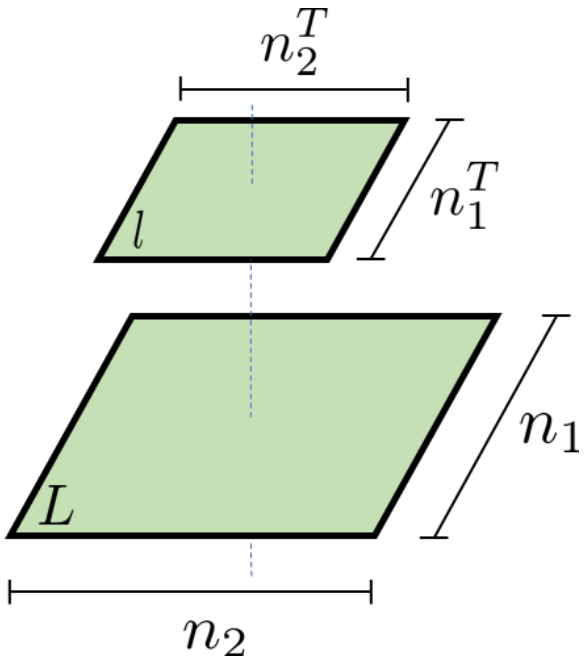


Figure 6.2: The large and small lattices, L and l.

is a stabilizer supported entirely on the first $n_1 n_2$ qubits, i.e. entirely on the large lattice. Let $G_1 \in \{0,1\}^{k_1 \times n_1}$ and $G_2 \in \{0,1\}^{k_2 \times n_2}$ be generator matrices for codes $\mathcal{C}_1$ and $\mathcal{C}_2$. Then, we can provide a generating set of stabilizers on the large lattice like

$$S_X^{(\text{hgp},L)} = (H_1 \otimes G_2)\,, \tag{6.23}$$

$$S_Z^{(\text{hgp},L)} = (G_1 \otimes H_2)\,. \tag{6.24}$$

Similarly, some stabilizers of HGP(H_1, H_2) are supported entirely on the small lattice. Let $F_1 \in \{0,1\}^{k_1^T \times n_1^T}$ and $F_2 \in \{0,1\}^{k_2^T \times n_2^T}$ be generating matrices for codes $\mathcal{C}_1^T$ and $\mathcal{C}_2^T$. The stabilizers on the small lattice have generating sets

$$S_x^{(\text{hgp},l)} = \left(F_1 \otimes H_2^T\right), \tag{6.25}$$

$$S_z^{(\text{hgp},l)} = \left(H_1^T \otimes F_2\right). \tag{6.26}$$

Logical operators of HGP(H_1, H_2) are those that commute with all stabilizers (we include the stabilizers themselves in this count). For instance, rows of the matrix $(I_{n_1} \otimes G_2, 0)$

indicate X-type logical operators, since

$$S_Z^{(\text{hgp})} \left(I_{n_1} \otimes G_2, 0\right)^T = 0. \tag{6.27}$$

The complete generating sets of X-type and Z-type logical operators are

$$L_X^{(\text{hgp})} = \begin{pmatrix} H_1 \otimes I_{n_2} & I_{n_1^T} \otimes H_2^T \\ I_{n_1} \otimes G_2 & 0 \\ 0 & F_1 \otimes I_{n_2^T} \end{pmatrix}, \tag{6.28}$$

$$L_Z^{(\text{hgp})} = \begin{pmatrix} I_{n_1} \otimes H_2 & H_1^T \otimes I_{n_2^T} \\ G_1 \otimes I_{n_2} & 0 \\ 0 & I_{n_1^T} \otimes F_2 \end{pmatrix}. \tag{6.29}$$

Nontrivial logical operators are logical operators that are not stabilizers.

An alternative representation of stabilizers and logical operators is to specify them by their supports. For instance $X^{(L)}(S)$ is an X-type Pauli supported on the qubits specified by $S \in \{0,1\}^{n_1 \times n_2}$ in the large lattice. Likewise for $X^{(l)}(T)$ with $T \in \{0,1\}^{n_1^T \times n_2^T}$ on the small lattice. Of course, Z-type Paulis $Z^{(L)}(S)$, $Z^{(l)}(T)$ are denoted analogously.

Using this support-matrix notation, we get alternative descriptions of the stabilizer groups

$$\mathcal{S}_X^{(\text{hgp})} = \big\{ X^{(L)}(S) X^{(l)}(T) : SH_2^T = H_1^T T, \tag{6.30}$$
$$G_1 S = 0, T F_2^T = 0 \big\},$$

$$\mathcal{S}_Z^{(\text{hgp})} = \big\{ Z^{(L)}(S) Z^{(l)}(T) : \; H_1 S = T H_2, \tag{6.31}$$
$$S G_2^T = 0, F_1 T = 0 \big\}.$$

and the logical operators

$$\mathcal{L}_X^{(\text{hgp})} = \{X^{(L)}(S)X^{(l)}(T) : SH_2^T = H_1^T T\}, \tag{6.32}$$

$$\mathcal{L}_Z^{(\text{hgp})} = \{Z^{(L)}(S)Z^{(l)}(T) : H_1 S = T H_2\}, \tag{6.33}$$

which are useful for discussing gauge-fixing later.

6.2.2 Subsystem Hypergraph Product Codes

In this section, we define the generalized Shor codes from [133] with notation similar to our description of HGP codes. This makes the two code families easier to relate later.

Definition 4. Let $H_1 \in \{0,1\}^{n_1^T \times n_1}$ and $H_2 \in \{0,1\}^{n_2^T \times n_2}$. The subsystem hypergraph product (SHP) code of H_1 and H_2 is the quantum subsystem code SHP(H_1, H_2) with gauge operators

$$G_X^{(\text{shp})} = (H_1 \otimes I_{n_2}), \tag{6.34}$$

$$G_Z^{(\text{shp})} = (I_{n_1} \otimes H_2). \tag{6.35}$$

It is worth noting that while the definition of HGP(H_1, H_2) depends on the parity check matrices H_1 and H_2, the definition of SHP(H_1, H_2) depends only on the codes $\mathcal{C}_1 = \ker H_1$ and $\mathcal{C}_2 = \ker H_2$. This is because the gauge groups $G_X^{(\text{shp})}$ and $G_Z^{(\text{shp})}$ are the same for SHP(H_1, H_2) and SHP(H_1', H_2') whenever row$(H_1) =$ row(H_1') and row$(H_2) =$ row(H_2').

Let us calculate the parameters $[\![N, K, D]\!]$ of the SHP code. There are clearly $N = n_1 n_2$ qubits in the code, which we place on a lattice like in Fig. 6.3.

To calculate K, begin by noticing that certain X-type operators, the bare X-type logical operators, commute with the entire group of gauge operators. These are generated by

$$L_X^{(\text{shp})} = (I_{n_1} \otimes G_2), \tag{6.36}$$

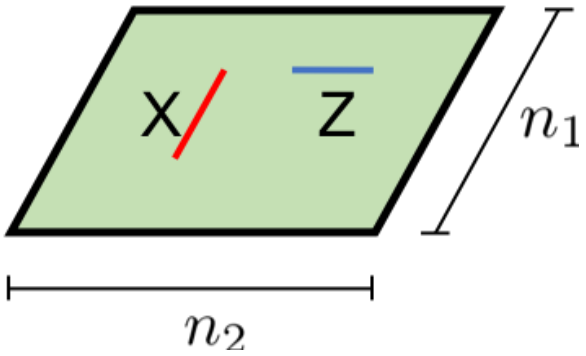


Figure 6.3: A subsystem hypergraph product code. For each column, X-type gauge operators are supported on qubits indicated by the parity checks H_1. For each row, Z-type gauge operators are supported on qubits indicated by the parity checks H_2.

because $L_X^{(\text{shp})} \left(G_Z^{(\text{shp})}\right)^T = 0$. Likewise, the bare Z-type logical operators are

$$L_Z^{(\text{shp})} = (G_1 \otimes I_{n_2}) \,. \tag{6.37}$$

The stabilizers of a subsystem code are those gauge operators that also commute with all elements of the gauge group, i.e. the center of the gauge group. These are generated by

$$S_X^{(\text{shp})} = (H_1 \otimes G_2) \,, \tag{6.38}$$

$$S_Z^{(\text{shp})} = (G_1 \otimes H_2) \,, \tag{6.39}$$

matching those stabilizers of HGP(H_1, H_2) that are supported entirely on the large lattice (see Eqs. (6.23), (6.24)).

Next, the number of encoded qubits can be calculated by comparing the ranks of $L_X^{(\text{shp})}$ and $S_X^{(\text{shp})}$ (or, equivalently of $L_Z^{(\text{shp})}$ and $S_Z^{(\text{shp})}$).

$$K = \text{rank}(L_X^{(\text{shp})}) - \text{rank}(S_X^{(\text{shp})}) \tag{6.40}$$

$$= n_1 k_2 - (n_1 - k_1) k_2 \tag{6.41}$$

$$= k_1 k_2 . \tag{6.42}$$

What does the description of SHP(H_1, H_2) look like in support-matrix notation? Writing down the relevant groups, we have

$$\mathcal{G}_X^{(\text{shp})} = \{X(S) : G_1 S = 0\}, \tag{6.43}$$

$$\mathcal{G}_Z^{(\text{shp})} = \{Z(S) : S G_2^T = 0\}, \tag{6.44}$$

$$\mathcal{L}_X^{(\text{shp})} = \{X(S) : S H_2^T = 0\}, \tag{6.45}$$

$$\mathcal{L}_Z^{(\text{shp})} = \{Z(S) : H_1 S = 0\}, \tag{6.46}$$

$$\mathcal{S}_X^{(\text{shp})} = \{X(S) : G_1 S = 0, S H_2^T = 0\}, \tag{6.47}$$

$$\mathcal{S}_Z^{(\text{shp})} = \{Z(S) : S G_2^T = 0, H_1 S = 0\}. \tag{6.48}$$

Dressed logical operators are denoted $\hat{\mathcal{L}}_X^{(\text{shp})} = \mathcal{L}_X^{(\text{shp})} \mathcal{G}_X^{(\text{shp})}$ and $\hat{\mathcal{L}}_Z^{(\text{shp})} = \mathcal{L}_Z^{(\text{shp})} \mathcal{G}_Z^{(\text{shp})}$.

To compute the distance D of the subsystem hypergraph product code, we need to find the minimum weight of an element of $\hat{\mathcal{L}}_X^{(\text{shp})} - \mathcal{G}_X^{(\text{shp})}$ or of $\hat{\mathcal{L}}_Z^{(\text{shp})} - \mathcal{G}_Z^{(\text{shp})}$. Let us suppose $M \in \hat{\mathcal{L}}_X^{(\text{shp})} - \mathcal{G}_X^{(\text{shp})}$. Then, M can be written as $M = X(S)X(T)$ where $X(S) \in \mathcal{L}_X^{(\text{shp})}$ and $X(T) \in \mathcal{G}_X^{(\text{shp})}$, so $S H_2^T = 0$ and $G_1 T = 0$. Also, since M is not in $\mathcal{G}_X^{(\text{shp})}$, there is some M' corresponding to a row of $L_Z^{(\text{shp})}$ that anticommutes with M. Glancing at Eq. (6.37), this means $M' = X(S')$ where S' is the outer product $S' = \vec{c}\,\hat{e}_j^T$ for some $\vec{c} \in \mathcal{C}_1$ and some j such that

$$\text{tr}(((S+T)^T S')) = \hat{e}_j^T (S+T)^T \vec{c} = 1. \tag{6.49}$$

This trace being 1 (modulo two) expresses the anticommutation of M and M'. Clearly, it implies $(S+T)^T \vec{c} \neq \vec{0}$. Because $\vec{c} \in \mathcal{C}_1$, there is a vector $\vec{x}$ such that $\vec{c} = G_1^T \vec{x}$ and accordingly,

$$(S+T)^T \vec{c} = S^T \vec{c} + T^T G_1^T \vec{x} = S^T \vec{c} \tag{6.50}$$

using $G_1 T = 0$. Moreover, $H_2 S^T \vec{c} = 0$ using $S H_2^T = 0$ and so $(S+T)^T \vec{c}$ is a nonzero vector in $\ker(H_2) = \mathcal{C}_2$. Thus, by definition of the classical code distance $|M| = |S+T| \geq |(S+T)^T \vec{c}| \geq d_2$.

Likewise, if we suppose $M \in \hat{\mathcal{L}}_Z^{(\text{shp})} - \mathcal{G}_Z^{(\text{shp})}$ we find $|M| \geq d_1$. Thus, we have shown $D \geq \min(d_1, d_2)$ and it is not hard given the form of $L_X^{(\text{shp})}$ and $L_Z^{(\text{shp})}$ to see that this in fact holds with equality $D = \min(d_1, d_2)$. Therefore, the subsystem hypergraph product code is a $[\![n_1 n_2, k_1 k_2, \min(d_1, d_2)]\!]$ code.

Quantum subsystem codes generalize quantum subspace codes because their stabilizers and logical qubits do not fix all the available degrees of freedom. The remaining degrees of freedom are counted as gauge qubits. These can be thought of as extra logical qubits that are not protected and thus not used to hold any meaningful information. If we calculate the number of gauge qubits in a subsystem hypergraph product code, we find it is

$$N - \text{rank}(S_X^{(\text{shp})}) - \text{rank}(S_Z^{(\text{shp})}) - K = (n_1 - k_1)(n_2 - k_2). \tag{6.51}$$

6.2.3 Example: A $[\![49, 16, 3]\!]$ Subsystem Hypergraph Product Code

Using the classical $[7, 4, 3]$ Hamming code for both the X and Z part, we can construct a $[\![49, 16, 3]\!]$ subsystem hypergraph product code by following Definition 4. We can construct a canonical set of logical operators of weight 3 and 4 for the 16 logical qubits encoded in the same code block, along with a set of 24 stabilizer generators. Note that similar to the BBS codes, for each of the stabilizers it suffices to use a single ancillary qubit to fault-tolerantly extract its syndrome, by performing CNOT gates in the order of gauge operators and hence directing propagated errors away from the direction of logical errors.

Similar to the $[\![21, 4, 3]\!]$ BBS code, we study the $[\![49, 16, 3]\!]$ SHP code under circuit level depolarizing noise as shown in Eq. 6.16. The results are shown in FIG. 6.4. Since the 4 encoded logical bits in the $[7, 4, 3]$ Hamming code have different performances and the constructed logical operators for the SHP code have different weights, the performance of the 16 encoded logical qubits varies and their pseudothreshold ranges between 10^{-4} and 2×10^{-4}.

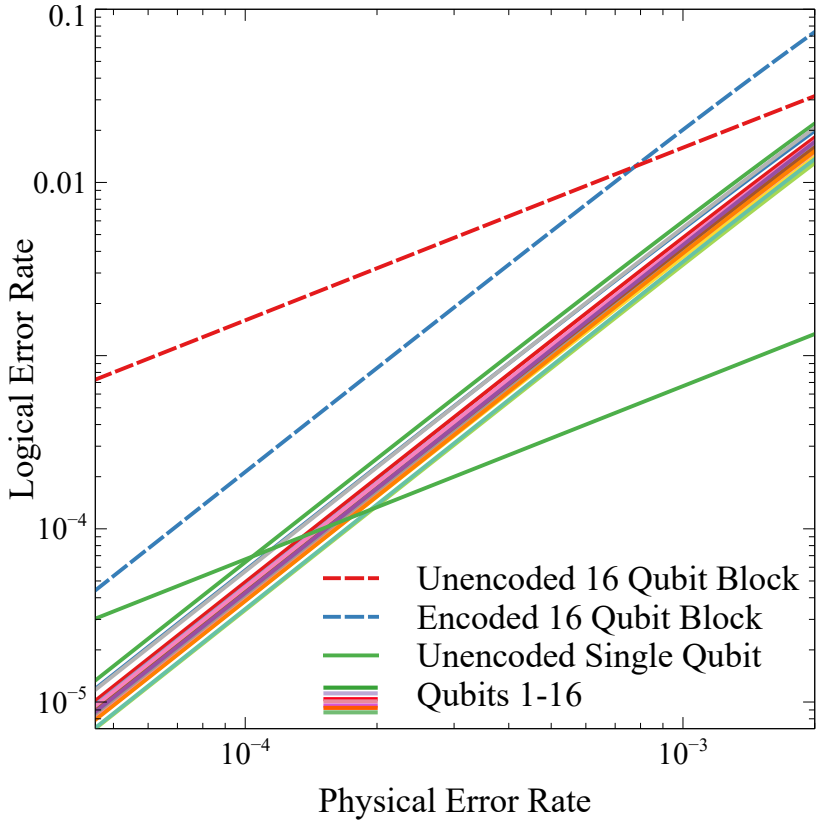


Figure 6.4: Simulated performance of the $[\![49, 16, 3]\!]$ SHP code under circuit level depolarizing noise, with one ancillary qubit per stabilizer for fault-tolerant syndrome extraction. The block pseudothreshold for the single code block with 16 encoded logical qubits is 8×10^{-4}, while the per logical qubit pseudothreshold ranges between 10^{-4} to 2×10^{-4}.

6.2.4 SHP Codes Gauge-Fix to HGP codes

To begin, we define gauge-fixing in general. See also [132]. We use the notation that for gauge group $\mathcal{G}$, its stabilizer group (centralizer) is $\mathcal{S}(\mathcal{G})$ and it encodes $K(\mathcal{G})$ qubits.

Definition 5. We say that the gauge group $\mathcal{G}'$ is a gauge-fixing of the gauge group $\mathcal{G}$ if

1. $\mathcal{S}(\mathcal{G}) \leq \mathcal{S}(\mathcal{G}') \leq \mathcal{G}' \leq \mathcal{G}$, and
2. $K(\mathcal{G}) = K(\mathcal{G}')$.

We also say that a code is a gauge-fixing of another code if their gauge groups are related in this way.

We noted below Eq. (6.39) that the stabilizers of SHP(H_1, H_2) are exactly those stabilizers of HGP(H_1, H_2) that are supported entirely on the large lattice. Similarly, one can check that the stabilizers of SHP(H_2^T, H_1^T) are those of HGP(H_1, H_2) supported entirely on the small lattice (i.e. Eqs. (6.25,6.26)). Also, Eq. (6.51) says that SHP(H_2^T, H_1^T) has $(n_2^T - k_2^T)(n_1^T - k_1^T)$ gauge qubits, which is the same number as SHP(H_1, H_2) by Eq. (6.19).

These two facts suggest the following theorem.

Theorem 6. *$\mathcal{Q}' =$ HGP(H_1, H_2) is a gauge-fixing of $\mathcal{Q} =$ SHP(H_1, H_2)SHP(H_2^T, H_1^T).*

Proof. We employ Definition 5. It should be clear that

$$K(\mathcal{Q}) = K(\text{SHP}(H_1, H_2)) + K(\text{SHP}(H_2^T, H_1^T)) \tag{6.52}$$

$$= k_1 k_2 + k_1^T k_2^T = K(\mathcal{Q}'), \tag{6.53}$$

therefore satisfying part (2) of the definition.

For part (1), it is important to associate (via a 1-1 map) the physical qubits of $\mathcal{Q}$ and $\mathcal{Q}'$. Recall, qubits of $\mathcal{Q}'$ are placed on the two lattices L and l. A qubit at site (i, j) in

SHP(H_1, H_2) is associated with the qubit at (i, j) in L. On the other hand, qubit (i, j) of SHP(H_2^T, H_1^T) is associated instead with the qubit at (j, i) on the small lattice l. Now, taken as a whole, this code $\mathcal{Q}$ has gauge operators and stabilizers that can be written as

$$\mathcal{G}_X^{(\mathcal{Q})} = \left\{ X^{(L)}(S) X^{(l)}(T) : G_1 S = 0, T F_2^T = 0 \right\}, \tag{6.54}$$

$$\mathcal{G}_Z^{(\mathcal{Q})} = \left\{ X^{(L)}(S) X^{(l)}(T) : S G_2^T = 0, F_1 T = 0 \right\}, \tag{6.55}$$

$$\mathcal{S}_X^{(\mathcal{Q})} = \{ X^{(L)}(S) X^{(l)}(T) : S H_2^T = 0, H_1^T T = 0, G_1 S = 0, T F_2^T = 0 \}, \tag{6.56}$$

$$\mathcal{S}_Z^{(\mathcal{Q})} = \{ Z^{(L)}(S) Z^{(l)}(T) : H_1 S = 0, T H_2 = 0, S G_2^T = 0, F_1 T = 0 \}. \tag{6.57}$$

It should now be clear that we have

$$\mathcal{S}_X^{\mathcal{Q}} \leq \mathcal{S}_X^{\text{hgp}} \leq \mathcal{G}_X^{\mathcal{Q}}, \tag{6.58}$$

$$\mathcal{S}_Z^{\mathcal{Q}} \leq \mathcal{S}_Z^{\text{hgp}} \leq \mathcal{G}_Z^{\mathcal{Q}}, \tag{6.59}$$

thereby satisfying part (1) of Def. 5. □

In essence, the two SHP codes live on the large and small lattices in Fig. 6.2, respectively, and gauge-fix to the HGP code by placing their gauge qubits in $(n_1 - k_1)(n_2 - k_2)$ maximally entangled two-qubit states.

6.3 Decoding BBS and SHP codes

Both the BBS codes and SHP codes can be decoded by directly running a classical decoder on the corresponding classical code used to construct the quantum code. In this section we review the decoding of BBS codes, as discussed in [132], and show that similar arguments can be applied to SHP codes. We review the classical belief propagation decoder for expander codes, and show how it can be used to tolerate measurement errors and therefore decode BBS and SHP codes.

6.3.1 Decoding the BBS Codes

To decode the BBS codes, we have to establish associations between the stabilizers of the quantum code and the parity checks of the classical code. For convenience, we assume that A is an $n \times n$ symmetric matrix constructed as $A = G^T Q G$, where G is the generating matrix of a $[n, k, d]$ classical code $\mathcal{C}$, so there is only one classical code $\mathcal{C} = \text{row}(A) = \text{col}(A)$ under consideration. We let H be the parity check matrix of $\mathcal{C}$.

Given Eq. 6.8, let S be the support of a X-type stabilizer of BBS(A), $X(S \cap A) \in \mathcal{S}_X^{(\text{bbs})}$. Since $SH_R^T = 0$, rows of S are codewords of $\mathcal{C}_R$, either all 1s or all 0s. Because $GS = 0$, columns of S are parity checks of $\mathcal{C}$. Therefore, $S \cap A = \text{diag}(\vec{r})A$ for some $\vec{r} \in \text{row}(H)$. Hence we have

$$\mathcal{S}_X^{(\text{bbs})} = \{X(\text{diag}(\vec{r})A) : \vec{r} \in \text{row}(H)\}. \tag{6.60}$$

Similarly,

$$\mathcal{S}_Z^{(\text{bbs})} = \{Z(A\text{diag}(\vec{c})) : \vec{c} \in \text{row}(H)\}. \tag{6.61}$$

Thus, the parity checks of the classical code indicate which sets of rows or columns constitute a stabilizer, and give us a one-to-one correspondence between the quantum stabilizers and the classical parity checks.

Since single qubit Pauli X errors within a column are equivalent up to gauge operators, each column is only sensitive to an odd number of Pauli X error. The even or oddness of a column corresponds to the 0 or 1 state of an effective classical bit in the code $\mathcal{C}$. Similarly, the symmetry of A indicates that the same correspondence holds for Pauli Z errors in rows and the even or oddness of rows in A.

Algorithm 1 (The Induced Decoder for BBS(A)). Given a symmetric binary matrix $A = G^T Q G$ where $\mathcal{C} = \text{row}(G) = \text{row}(A) = \text{col}(A)$ is a classical $[n, k, d]$ code, we can decode the Bravyi-Bacon-Shor code BBS(A) by:

- Collect the X- or Z-type syndrome $\vec{\sigma}$ for the quantum code BBS(A).
- Run the classical decoder to obtain a set of corrections for the classical code $\vec{c} = \mathcal{D}(\vec{\sigma})$.
- For each bit in the correction $\vec{c}$, apply a Pauli Z- or X-type correction to a single qubit in each row or column corresponding to the classical bit.

The time complexity of the induced decoder consists of the time to construct the stabilizer values and the time to run the classical decoder $\mathcal{D}$. Given an $[n, k, d]$ classical code $\mathcal{C}$, for a weight-w parity check of the classical code the corresponding stabilizer of the BBS(A) code is the sum of $O(wn)$ two-qubit gauge measurement. There are m such stabilizers and suppose the classical decoder runs in time at most t, then the induced decoder of BBS(A) takes time $O(mwn + t)$. When classical expander codes are used to construct BBS(A) and the belief propagation decoder is used as classical decoder $\mathcal{D}$, $m = O(n)$, $w = O(1)$, $t = O(n)$, so the induced decoder runs in time $O(mwn + t) = O(n^2 + n) = O(N)$, which is linear in the size of the quantum code.

6.3.2 Decoding the SHP Codes

Similar to what we have done for the BBS codes, to decode the SHP codes we have to associate the stabilizers of the quantum code to the parity checks of the classical codes. To illustrate the idea most easily, we assume that the X and Z part of the SHP code are generated by the same $[n, k, d]$ classical code $\mathcal{C}$ with generating matrix $G \in \mathcal{F}_2^{k\times n}$ and parity check matrix $H \in \mathcal{F}_2^{m\times n}$, SHP($H_1, H_2$) = SHP($H$). From Eq. (6.38) and Eq. (6.39) we have

$$S_Z^{(\text{shp})} = Z(G \otimes H) \tag{6.62}$$

$$= \{Z(g^T \otimes h) : g \in \text{row}(G), h \in \text{row}(H)\}, \tag{6.63}$$

$$S_X^{(\text{shp})} = X(H \otimes G) \tag{6.64}$$

$$= \{X(h^T \otimes g) : h \in \text{row}(H), g \in \text{row}(G)\}. \tag{6.65}$$

Since rank$(G) = k$, the eigenvalues of the quantum stabilizers correspond to exactly k sets of syndromes for the classical code $\mathcal{C}$. In the case of Z-type stabilizers, let g_i be the i-th row of G for each i such that $1 \leq i \leq k$. A set of syndromes for the classical code $\mathcal{C}$ is generated by measuring the following set of stabilizers

$$\{Z(g_i^T \otimes h_j) : h_j \in \text{row}(H), 1 \leq j \leq m\}. \tag{6.66}$$

These k sets of syndromes are passed to the classical decoder and results in k sets of n-bit corrections on $\mathcal{C}$. However, in order to apply these k sets of classical corrections canonically onto independent sets of qubits in the SHP code without affecting each other, we have to make sure that the generating matrix G is in the reduced row echelon form, so that the i-th set of corrections can be applied on the i-th row of qubits lattice.

Algorithm 2 (**The Induced Decoder of SHP**(H))**.** Given a $[n, k, d]$ classical code $\mathcal{C}$ with generating matrix $G \in \mathcal{F}_2^{k,n}$ and parity check matrix $H \in \mathcal{F}_2^{m,n}$, the hypergraph subsystem code SHP(H) can be decoded by

- Reshape G into its reduced row echelon form
  $G = [I_k \ B]$.
- Collect the X- or Z-type syndrome $\vec{\sigma}$ of the quantum code SHP(H).
- For each $i \in \{1, 2, \ldots, k\}$, the syndrome corresponding to the set of stabilizers $\{Z(g_i^T \otimes h_j) : h_j \in \text{row}(H), 1 \leq j \leq m\}$ is passed to the classical decoder $\mathcal{D}$, and a n-bit correction $\vec{c_i}$ is obtained.
- For each set of corrections $\vec{c_i}$, Pauli Z- or X-type corrections are applied to the qubits $\vec{e_i} \otimes [1, 1, \ldots, 1]$, where $\vec{e_i}$ is the i-th unit vector.

Hence when using the induced decoder on the quantum code SHP(H) that is constructed by a $[n, k, d]$ code, we have to run the classical decoder $\mathcal{D}$ a total of k times. The correction consists of k sets of n-qubit Paulis have to be applied on the first k rows or columns of the qubit lattice.

The time complexity of the induced decoder for SHP codes again consists of the time to construct the stabilizer values and the time to run the classical decoder $\mathcal{D}$. For a weight-w parity check of the classical code, the corresponding stabilizer of the SHP code is the sum of $O(n)$ number w-qubit gauge measurements. There are $O(k \times m)$ stabilizers, and the classical decoder $\mathcal{D}$ needs to be run k times where each run takes time at most t, then the induced decoder takes time $O(kmnw + kt)$. When classical expander codes are used to construct the SHP code and the belief propagation decoder is used as the classical decoder $\mathcal{D}$, $m = O(n)$, $k = O(n)$, $w = O(1)$, $t = O(n)$, so the induced decoder runs in time $O(kmnw + kt) = O(n^3 + nt) = O(N^{3/2})$.

6.3.3 Classical Belief Propagation Decoder

In the previous two sections we have shown that decoding both the BBS codes and the SHP codes amount to directly decoding the underlying classical code $\mathcal{C}$ that was used to construct the quantum code, and apply the resulting corrections to the appropriate set of qubits in the quantum code. Therefore, in order to maximize the performance of the induced decoding algorithm the best classical decoder should be employed with modifications to tolerate measurement noise. Sipser and Spielman have analyzed the flip decoder [129, 134] for classical expander codes and in the scenario that the parity checks are noisy in addition to the bits. A quantum version of the classical flip decoder has been shown to decode the quantum expander codes efficiently [126, 135, 127, 126].

However, when classical LDPC codes and expander codes are considered, various iterative message-passing decoding algorithms have been shown to result in codes with rate approaching the Shannon capacity together with efficient decoding algorithm (see e.g. [136]).

Message passing algorithms get the name as information is transmitted back and forth between variable and check nodes along the edges of the graph that is used to define the classical code. The transmitted message along an edge is a function of all received messages at the node except for a particular edge. This property ensures that the incoming messages are independent for a tree like graph. Among these well-known decoding algorithms, the belief propagation (BP) decoder, sometimes referred to as Gallager's soft decoding algorithm [20], have been shown to out perform other message-passing algorithms for classical LDPC codes when the binary symmetric channel (BSC) is considered. In this section we briefly describe the BP decoder for classical LDPC codes. For a comprehensive discussion of this area, we point the reader to the book by Richardson and Urbanke [21] and the notes by Guruswami [137], which are excellent resources on this topic.

In particular, here we present the modified BP decoder that uses parity check values as input instead of bit values, in order to simulate the quantum case where data qubit values are not known to decoders. In order to run the BP decoder using parity check values, we add another set of m "syndrome nodes" $s_j, 1 \leq j \leq m$, that have one-to-one correspondence to the check nodes: syndrome node s_j and check node j are connected by edge (s_j, j). These syndrome nodes s_i are used to store the measured parity check values. Without loss of generality, we assume that the all 0s message is the correct message to be received.

Algorithm 3 (Belief Propagation Decoding Algorithm). Assuming the probability p for each bit of the incoming message to be flipped is the same, then the log-likelihood ratio m_i of the i-th bit is

$$m_i = \log \frac{1-p}{p}. \tag{6.67}$$

For the syndrome nodes s_i, we let $m_{s_i} = +\infty$ if the i-th syndrome is 0 and $m_{s_i} = -\infty$ if the i-th syndrome is 1. Do the following two steps alternatively:

1. **Rightbound messages**: For all edges $e = (i, j)$, $i \in \{1, 2, \ldots, n\} \cup \{s_1, s_2, \ldots, s_m\}$, do the following:

- if this is the zeroth round, $g_{i,j} = m_i$.
- Otherwise

$$g_{i,j} = m_i + \sum_{k \in \mathcal{N}(i) \setminus j} h_{i,k} \tag{6.68}$$

where $\mathcal{N}(i)$ denotes the set of neighbors of node i.

The variable node i sends the message $g_{i,j}$ to check node j.

2. **Leftbound messages**: For edges $e = (i, j)$, $i \in \{1, 2, \ldots, n\}$ do the following:

$$h_{i,j} = f\left(\prod_{k \in \mathcal{N}(j) \setminus i} \frac{e^{g_{k,j}} - 1}{e^{g_{k,j}} + 1}\right), \quad f(u) = \log \frac{1+u}{1-u}. \tag{6.69}$$

The check node j sends the message $h_{i,j}$ to node i.

At each step we can determine the current variable node values v_i given their updated log-likelihood ratios: $v_i = 0$ if $m_i > 0$ and $v_i = 1$ if $m_i < 0$. The above iterative step terminates when all check nodes are satisfied based on the current v_i, or the predetermined number of iterations is reached. The variable node value v_i at the final step is used as correction for the noisy channel output b_i.

If the graph considered has large enough girth when compared to the number of iterations of the algorithm, the messages at each iteration would approach the true log-likelihood ratio of the bits given the observed values. By applying expander graph arguments to message passing algorithms it has been shown that the BP decoding algorithm can correct errors efficiently, with time linear in the block size [138]. Therefore the belief propagation decoder is a good candidate for decoding the BBS and SHP codes constructed using classical LDPC codes.

6.3.4 Handling Measurement Errors with BP Decoder

As we mentioned previously, decoding algorithms for classical codes usually do not consider the problem of measurement noise. In previous studies of the BP decoder, no explicit proposals have been made regarding handling measurement noise when decoding classical expander codes. In order to use the BP decoder to decode the BBS and SHP codes as part of the induced decoder, modifications have to be made in order to tolerate measurement errors on parity check measurements.

When given a classical code $\mathcal{C}$ with n variable nodes and m check nodes, in addition to what we have done in Algorithm 3 we add m variable nodes to the graph so that each of them has a one-to-one correspondence with the m check nodes: variable node $n+j$ is connected to check node j via edge $(n+j, j)$. These additional variable nodes are used to represent measurement errors on the parity checks. An example of the modified graph for decoding a classical linear code of block length 6 is shown in FIG. 6.5. In the binary symmetric channel, let p be the probability that a bit is flipped and let q be the probability that a measurement is flipped. We define the log-likelihood ratio m_i for the $n+m$ variable nodes as:

- $m_i = \log \frac{1-p}{p}, 1 \leq i \leq n$,
- $m_i = \log \frac{1-q}{q}, n+1 \leq i \leq n+m$.

For the syndrome nodes s_i, we let $m_{s_i} = +\infty$ if the i-th syndrome is 0 and $m_{s_i} = -\infty$ if the i-th syndrome is 1.

When executing the belief propagation decoding algorithm, the normal message passing process is executed as described in Algorithm 3. For the added syndrome node s_i, they send their log-likelihood ratios m_i to the associated check node j with message $g_{i,j} = m_i$ during the rightbound messages phase in each iteration, but there will be no incoming messages from the check nodes c_i to change their values. The algorithm terminates when all

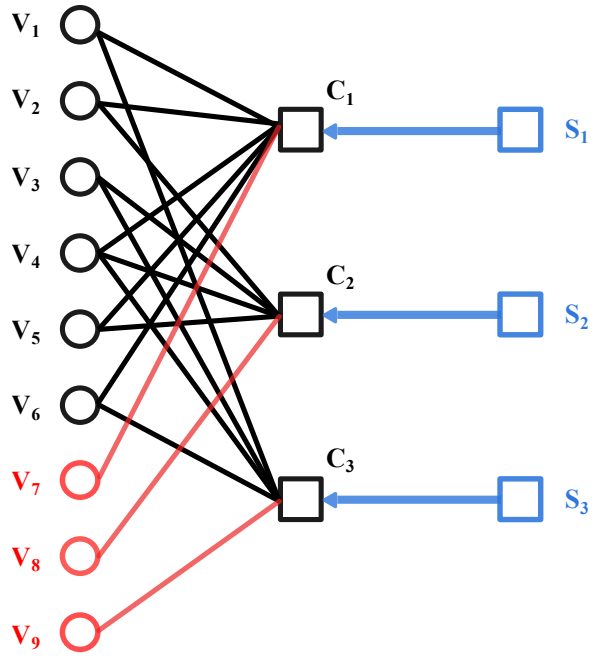


Figure 6.5: The graph for decoding a classical code of length 6 using the modified BP decoder that tolerates measurement errors. The syndrome nodes s_1, s_2, s_3 are assigned log-likelihood values $\pm\infty$ given the input parity check measurement values 0 or 1.

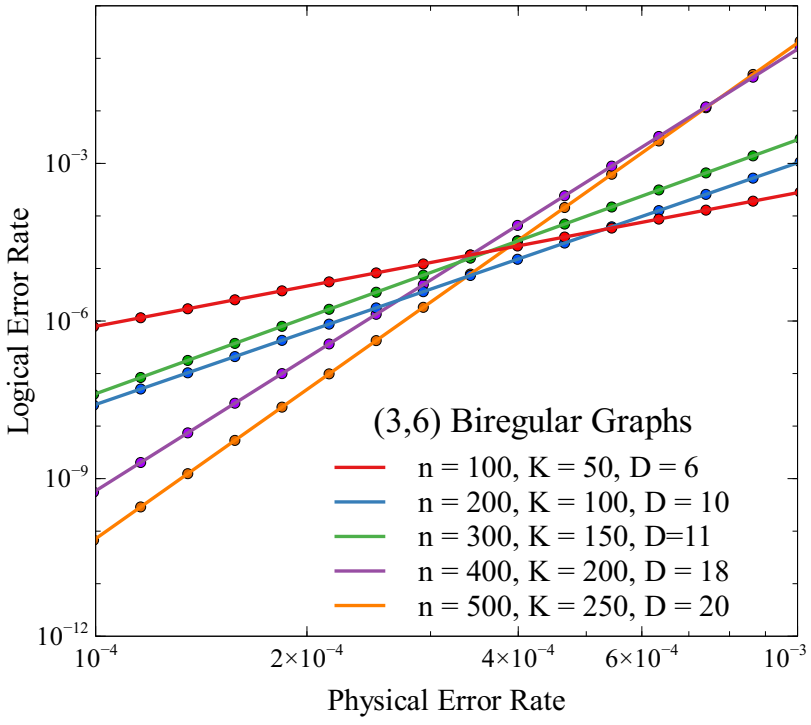
check nodes are satisfied or a predetermined number of iterations is reached, and the n-bit variable node values at the final step are used as corrections for the noisy data qubits.

By employing the above described modifications to the BP algorithm, we can efficiently decode the classical expander codes while tolerating measurement errors.

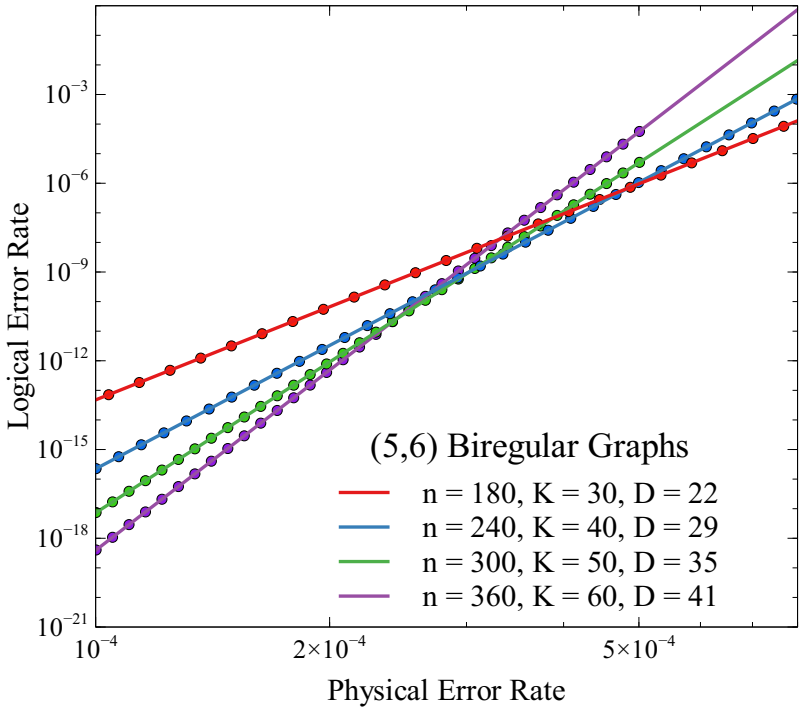
6.4 Numerical Simulations and Results

In this section we present numerical results of decoding the BBS and SHP codes using the induced decoders instantiated with the modified BP decoder that handles measurement errors. All simulations are done under the phenomenological error model, where given probability p, random single-qubit bit or phase flip errors of the form $E_{1q} = \{\sqrt{1-p}I, \sqrt{p}X\}$ or $E_{1q} = \{\sqrt{1-p}I, \sqrt{p}Z\}$ are applied independently on qubits and measurements output the wrong (opposite) value with probability p. There is no circuit-level error propagation in the simulation.

In order to maximize the parameters and performance of the quantum codes when de-



(a)



(b)

Figure 6.6: Simulating the performance of Bravyi-Bacon-Shor codes constructed using (a) $(3, 6)$- and (b) $(5, 6)$-biregular bipartite graphs. BBS codes by $(5, 6)$ graphs outperforms BBS codes by $(3, 6)$ graphs due to superior performance of classical $(5, 6)$ codes.

coded by the induced decoders, we construct the BBS and SHP codes with classical regular LDPC codes defined by biregular bipartite graphs. To obtain symmetric performance for X- and Z-type errors, both X and Z part of each quantum code are constructed with the same classical LDPC code. Since both the BBS and SHP codes are defined as CSS codes, X- and Z-type errors can be decoded separately using the induced decoder. Hence in the rest of the chapter we assume that each qubit independently suffers from Pauli X- and Z-type errors as described in the previous paragraph, and study the performance of these codes by plotting the average logical error rate per logical qubit of the K-qubit block versus the physical error rate of each qubit. Using this metric allows us to directly compare the average performance of quantum codes with different encoding rates on an equal footing, instead of comparing large blocks with vastly different numbers of encoded qubits. By doing so we are taking into account both the performance and encoding rate when comparing different codes, but to some extent ignoring the potential correlation between logical errors.

The classical regular LDPC codes that are used to construct the BBS and SHP codes were randomly generated biregular bipartite graphs using the configuration model [21]. It can be shown that asymptotically these graphs will have a good expansion coefficient, making them classical expander codes with good performance. For each of the selected block size, we randomly generated 1000 biregular bipartite graphs with specified node degrees and simulated their performance under the binary symmetric channel. The best-performing classical code is chosen to construct the quantum code. Since the induced decoder for the quantum code directly decodes on the underlying classical code, a relatively good classical code implies a relatively good quantum code.

We studied two classes of graphs for generating classical LDPC codes: the $(3, 6)$- and $(5, 6)$- biregular bipartite graphs, which we will refer to as the $(3, 6)$ and $(5, 6)$ codes. By simulating the performance of these two classical codes with the BP decoder, we observed that the $(5, 6)$ codes significantly outperform the $(3, 6)$ codes, which agrees with previous studies in classical coding theory [130, 139]. Given a (b, c) code of size n, the number

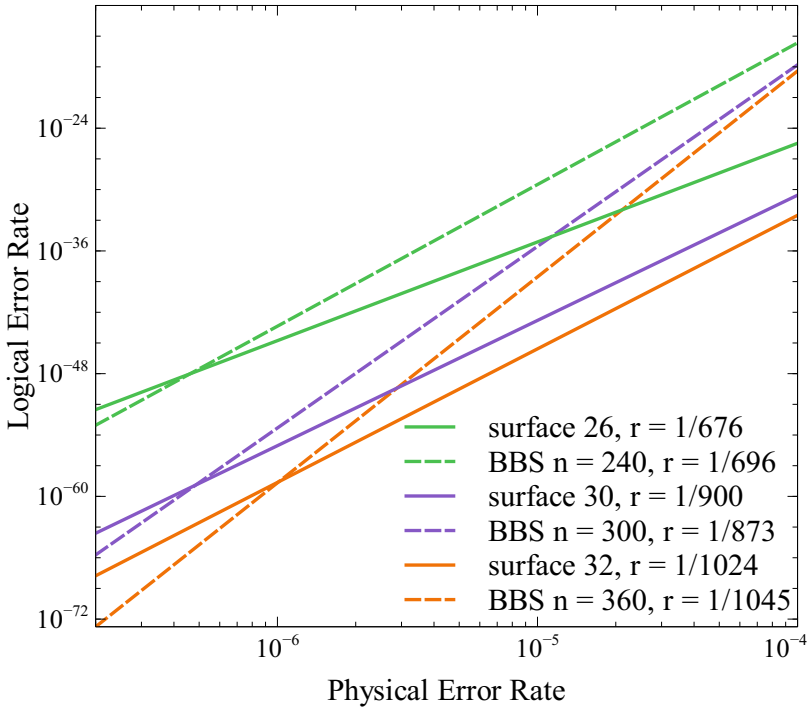


Figure 6.7: Comparing the average error rate per logical qubit of the BBS codes constructed with (5,6)-biregular bipartite graphs of block size $240, 300, 360$ to surface codes of sizes $26 \times 26, 30 \times 30, 32 \times 32$.

of encoded bits is $k = \frac{c-b}{b}n$ and the encoding rate for the classical code is $\frac{c-b}{b}$. Hence the BBS codes constructed with (b, c) classical code have parameters $[\![N_{BBS}, K_{BBS}]\!] = [\![O(n^2), \frac{c-b}{b}n]\!]$, and the SHP codes have parameters $[\![N_{SHP}, K_{SHP}]\!] = [\![n^2, (\frac{c-b}{b})^2 n^2]\!]$.

In all plots, n is the number of bits/variable nodes for the classical LDPC code, N is the number of physical qubits in the quantum code, K is the number of encoded logical qubits in the quantum code, D is the average distance of the quantum code found through fitting the simulated data to $P_L = Ap^D$, and r is the encoding rate of the quantum code. The numerical performance of the BBS codes presented in Figures 6.6, 6.7 and 6.9 are obtained using importance sampling to error rates as low as 10^{-4}, and best-fit lines are plotted in order to extrapolate the codes behavior to low error regimes. Details of importance sampling can be found in [22]. The numerical performance of the SHP codes presented in Figures 6.8 and 6.9 are obtained through Monte Carlo simulations at various physical error rates.

From FIG. 6.6 we can see that the BBS codes constructed with $(5, 6)$ codes have sig-

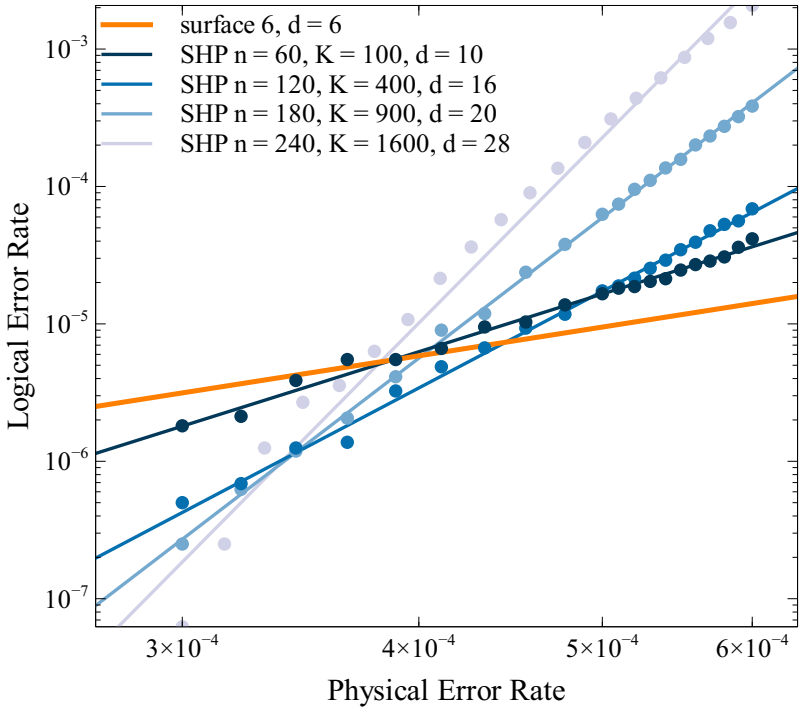


Figure 6.8: Simulating the performance of the SHP codes constructed using $(5, 6)$-biregular bipartite graphs. Their average performance per logical qubit are compared to the size 6×6 surface code. All codes in this plot have encoding rate $1/36$.

nificantly better performance than that with $(3, 6)$ codes, as expected given the results on the classical codes. It is clear from FIG. 6.6 that the BBS codes do not have a fault-tolerant threshold, due to the fact that the weight-2 gauge operators in the quantum code result in a superexponential scaling of number of weight-D dressed logical operators. A similar behavior is observed for the SHP codes constructed with $(5, 6)$ codes, as shown in 6.8, where the SHP codes also do not exhibit a fault-tolerant threshold.

To benchmark the performance of the BBS codes and SHP codes, we compare them to the surface codes as well as to each other. As previously mentioned, in order to obtain a reasonable comparison we compare the BBS and SHP codes against surface codes of similar encoding rate r by comparing the average error rate of single logical qubits within the same code block to the logical error rate of the surface code. In FIG. 6.7 we are comparing the average logical error rate per logical qubit of the BBS codes constructed with $(5, 6)$ codes of sizes $n = 240, 300, 360$ to surface codes of sizes $26 \times 26, 30 \times 30, 32 \times 32$. Note

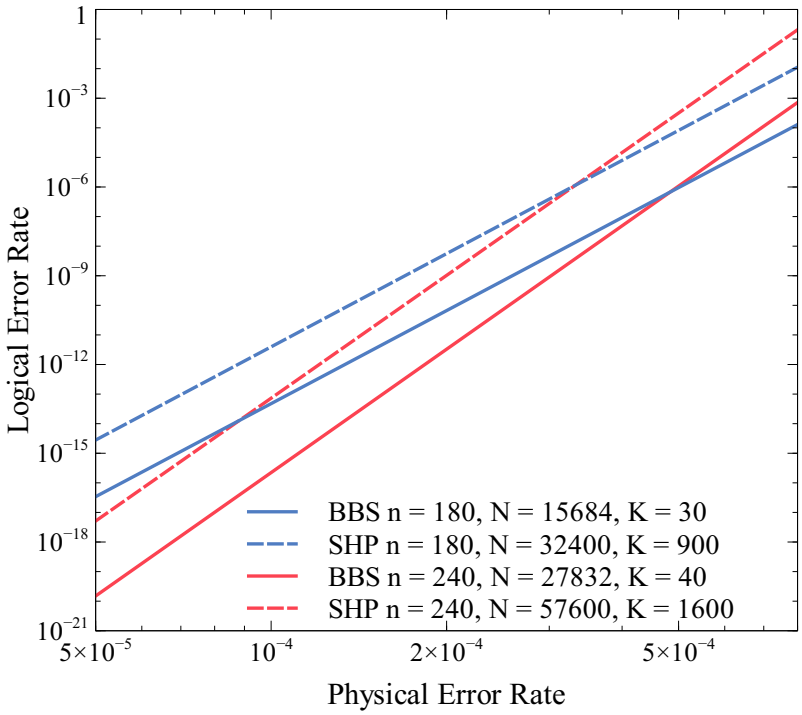


Figure 6.9: Comparing the average performance per logical qubit of the BBS codes to the SHP codes. The BBS and SHP codes with the same n are constructed using the same $(5, 6)$-biregular bipartite graph.

that the surface code results are simulated using the Union Find decoder [140], so that we are comparing a linear time decoding algorithm of the BBS codes to a linear time decoding algorithm of the surface code. The BBS codes have better distances than surface codes of similar encoding rates, but they only outperform the surface codes for physical error rates below 10^{-6}. Similar results for SHP codes are shown in FIG.6.8. Since the SHP codes have constant encoding rates and when the $(5, 6)$ codes are used, the resulting encoding rate is $r = 1/36$, so we are comparing the average error rate for single logical qubits in the SHP codes to a single 6×6 block of surface code. The SHP codes can have significantly better distance than surface code of the same encoding rate, but they do not outperform the surface code until physical error rates $p \leq 4 \times 10^{-4}$.

Finally, we compare the average performance per logical qubit of the BBS and SHP codes, as shown in 6.9. The comparison is made between BBS and SHP codes constructed using the exact same $(5, 6)$-biregular bipartite graph. While it seems that the SHP codes'

average logical qubit performance is slightly worse than that of the BBS codes, bear in mind that the SHP codes have much higher encoding rate.

6.5 Conclusion

When constructed using classical expander codes, the BBS codes have encoding rates $O(1/\sqrt{N})$ and the SHP codes have constant encoding rates dependent on the expander code parameters. Suppose the same classical expander code is used, the resulting SHP codes have even higher encoding rates than the hypergraph product codes. Hence for large block sizes these codes could offer significant savings in terms of resource overhead when trying to achieve a specific logical error rate. It is worth noting that while we are already observing very good logical performance by simulating codes constructed with small biregular bipartite graphs, classical LDPC codes asymptotically become better expander codes and the belief propagation decoder will give a much better performance for expander codes of larger block sizes.

CHAPTER 7
CONCLUSION

In this thesis, we have explored several directions of study in the field of quantum error correction with different perspectives and constraints in mind. While all of the results presented here belong to the regime of theoretical studies and abstract modeling, our motivation always comes from practical concerns in actual experimental systems: the fragility of quantum information due to interaction with the environment, the error-proneness of individual quantum operations, and the immense resource overhead for realizing fault-tolerant quantum computing. While our studies here mainly focused on the trapped ion quantum computing architecture, it is worth noting that the above mentioned concerns exist regardless of which technology and platform is being used as physical implementation of the quantum system. Therefore, although the choice of error correcting codes and specific methods of implementing quantum operations can vary, it remains essential to study the nature of errors present in these physical systems and find ways to efficiently suppress noise when realistic constraints are imposed. Towards this goal, we have contributed the following.

7.1 Fault-Tolerance on Near-Term Quantum Devices

In Chapter 3, we presented properties of a new $[[7, 1, 3]]$ stabilizer code that can achieve fault-tolerant syndrome measurements using a single ancillary qubit under the anisotropic error model, and showed why certain two-qubit errors under the standard depolarizing error model would prevent this code from achieving fault-tolerant measurements. In particular, the limit on total number of syndrome outcomes makes it impossible for the lookup table decoder to detect and correct all two-qubit errors in the standard depolarizing error model. Additionally, we showed that using the flag method the Bare $[[7, 1, 3]]$ code can

achieve fault-tolerance under both the standard depolarizing model and the anisotropic model. Overall, with the flag method the Bare $[[7, 1, 3]]$ code shows better performance under realistic error models.

In order to gain a thorough understanding of near-term error correction experiments in the trapped ion architecture, in [55] we explored the feasibility of implementing the Surface-17 quantum error correcting code, which is a popular choice of implementation for near-term quantum error correction demonstration, using a linear chain of $^{171}\mathrm{Yb}^{+}$ ions. Measurement of the ion state by fluorescence requires that the ancilla qubits be physically separated from the data qubits using shuttling to avoid errors due to scattered photons. We minimized the time required to execute the stabilizer measurement circuit by optimizing the mapping of the two-dimensional surface code to the linear chain of ions. We developed a physically motivated Pauli error model that allows for fast simulation and captures the key sources of noise in an ion trap quantum computer including gate imperfections and ion heating. Our numerical simulation showed a consistent requirement of two-qubit gate fidelity $\geq 99.9\%$ for the logical memory to have a better fidelity than physical two-qubit operations.

In Chapter 4, we found that for subsystem and subspace quantum error correcting codes derived from the compass model, the gauge operators prescribe a method for efficient fault-tolerant stabilizer measurement. This result yields the ability to fault-tolerantly measure large stabilizer operators with single ancillary qubits and does not require any additional resource overhead when the required qubit connectivity is provided. We compared the performance of Bacon-Shor-13 using this new method to Surface-17, and found that Bacon-Shor-13 outperforms Surface-17 in all measures for a simple circuit: ion trap operation time, logical error rate, and number of qubits required. The key advantage of Bacon-Shor-13 over Surface-17 comes from its greatly simplified state preparation. In addition, the lower qubit count makes Bacon-Shor-13 a more immediate target for near-term quantum error correction in systems where non-nearest neighbor gates are possible, such as trapped

ions. However as Surface-17 holds advantage over Bacon-Shor-13 in terms of error correction, multiple rounds of error correction will begin to favor Surface-17.

In Chapter 5, we showed that there are a wide variety of optimal codes when considering different error sources, indicating the importance of being able to accurately benchmark a system and find the error models and parameters which describe it. When benchmarked using the logical *CNOT* circuit, in the depolarizing error model Surface-17 clearly outperforms all other codes, however when considering physically realistic error models Bacon-Shor-13 and Shor's code variants perform better. Due to the all-to-all connectivity present in ion trap systems, the surface code does not benefit significantly from its locality. We also provided evidence for the damaging effects of crosstalk errors in error correction systems, and proposed solutions to mitigate these errors when specific error correcting codes are considered.

To further pursue our investigation on damaging effects of coherent errors and ways to mitigate them, we developed stabilizer slicing [112], which is a new and simple technique for suppressing coherent erorrs in syndrome extraction circuits. It requires certain experimental capacities but no additional overhead, and dramatically improves the logical fidelity of syndrome extraction with the same-quality physical gates. Because it requires no additional resources, we hope that even its 2-body iteration could yield significant benefit in realistic near-term fault-tolerance experiments where systematic coherent error is a dominant factor.

7.2 Subsystem Quantum Error Correcting Codes

In Chapter 6, We studied two different constructions of quantum subsystem error-correcting codes using classical linear codes: the Bravyi-Bacon-Shor (BBS) codes and the subsystem hypergraph product (SHP) codes. We reviewed the BBS codes that was introduced in a previous paper [132], and presented a construction of the SHP codes that can be viewed similar to the hypergraph product codes [125]. We proposed efficient algorithms to decode

the BBS and SHP codes while handling measurement errors by using a modified belief propagation decoder for classical expander codes. We studied the numerical performance of the BBS and SHP codes, and showed that while these codes do not have a fault-tolerant threshold, they have very good distance scaling and encoding rates.

Therefore, the BBS and SHP codes are worth studying for the purpose of large scale quantum error correction. Future studies on these codes could include investigating the potential of using large irregular LDPC codes to construct the BBS and SHP codes in order to achieve better logical performances, tailoring the quantum code for biased noise models by using two different classical codes to construct asymmetric BBS and SHP codes, and methods to apply fault-tolerant logical operations within the same code block.

To extend our findings in [44] to large distance quantum codes, in [63] we have described an ansatz for designing planar codes stemming from the 2-D compass model. We have provided evidence that simple subfamilies of this class may be useful for correcting biased noise in idealized code capacity and phenomenological noise models, particularly if that bias is distributed geometrically. In particular, one can bias the stabilizers locally towards correcting a certain error-type. There are two central challenges for these codes in the more realistic circuit-level noise model. Although these codes are still local, there is a trade-off between the bias of the codes and the locality of the stabilizer measurements. We have demonstrated that fault-tolerant measurement in Bacon-Shor [30, 44] and surface codes [50, 31] using bare ancilla can be adapted to the compass model, if measurements are performed in the correct order. Nevertheless, these correlated errors will deteriorate code performance as higher-weight stabilizer outcomes become less reliable. This might be mitigated by using other flag-type schemes, or by preserving some gauge degrees of freedom. We would expect that these gains would persist, but at the expense of higher bias and code overhead. As such, we leave a more involved circuit-level analysis to future work. The second concern is whether the biased noise model itself can persist at the circuit level. To remain experimentally motivated, one must choose operations that preserve

the bias [141, 142, 143]. Consequently, the construction of *simple* and *bias-preserving* fault-tolerant gadgets is key to utilizing asymmetric noise. Finally, we have only narrowly broached the design space offered by these codes. Exploring different configurations according to other geometrically-defined noise [140], generalizing to codes defined on the 3-D compass model, and using correlated decoders [144, 145, 142, 146, 147] are all avenues to explore. More generally, finding other low-density parity check constructions adapted to biased noise may give the best of both worlds, mitigating the overhead of asymmetrization while taking advantage of the bias.

7.3 Final Remarks

It has been almost three decades since the idea that quantum computers have the potential to outperform classical computers was first proposed. During this time, research around quantum computation has accelerated and this field has evolved from theoretical physics into a conglomerate of physics, mathematics, computer science, and engineering. Although it has not yet been experimentally demonstrated that practical large-scale fault-tolerant quantum computers exist, it is my personal belief that the multidisciplinary research effort on quantum computers will eventually take us there. In the meantime, fruitful results along the journey would not only benefit the quantum computation community but also provide positive impact on related research fields.

REFERENCES

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. United Kingdom: Cambridge University Press, 2000.

[2] M.-D. Choi, "Completely positive linear maps on complex matrices," *Linear algebra and its applications*, vol. 10, no. 3, pp. 285–290, 1975.

[3] D. Gottesman, "The heisenberg representation of quantum computers, talk at," in *International Conference on Group Theoretic Methods in Physics*, Citeseer, 1998.

[4] Y. Shi, "Both toffoli and controlled-not need little help to do universal quantum computing," *Quantum Information & Computation*, vol. 3, no. 1, pp. 84–92, 2003.

[5] C Monroe, R Raussendorf, A Ruthven, K. Brown, P Maunz, L.-M. Duan, and J Kim, "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects," *Physical Review A*, vol. 89, no. 2, p. 022 317, 2014.

[6] A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. Poschinger, C. Hempel, J Home, *et al.*, "Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation," *Physical Review X*, vol. 7, no. 4, p. 041 061, 2017.

[7] E. Knill, R. Laflamme, and G. J. Milburn, "A scheme for efficient quantum computation with linear optics," *nature*, vol. 409, no. 6816, pp. 46–52, 2001.

[8] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, and G. J. Milburn, "Linear optical quantum computing with photonic qubits," *Reviews of Modern Physics*, vol. 79, no. 1, p. 135, 2007.

[9] J. Clarke and F. K. Wilhelm, "Superconducting quantum bits," *Nature*, vol. 453, no. 7198, pp. 1031–1042, 2008.

[10] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "A quantum engineer's guide to superconducting qubits," *Applied Physics Reviews*, vol. 6, no. 2, p. 021 318, 2019.

[11] J. Weber, W. Koehl, J. Varley, A Janotti, B. Buckley, C. Van de Walle, and D. D. Awschalom, "Quantum computing with defects," *Proceedings of the National Academy of Sciences*, vol. 107, no. 19, pp. 8513–8518, 2010.

[12] L. Childress and R. Hanson, "Diamond nv centers for quantum computing and quantum networks," *MRS bulletin*, vol. 38, no. 2, pp. 134–138, 2013.

[13] D. Loss and D. P. DiVincenzo, "Quantum computation with quantum dots," *Physical Review A*, vol. 57, no. 1, p. 120, 1998.

[14] A Imamog, D. D. Awschalom, G. Burkard, D. P. DiVincenzo, D. Loss, M Sherwin, A Small, *et al.*, "Quantum information processing using quantum dot spins and cavity qed," *Physical review letters*, vol. 83, no. 20, p. 4204, 1999.

[15] H.-J. Briegel, T. Calarco, D. Jaksch, J. I. Cirac, and P. Zoller, "Quantum computing with neutral atoms," *Journal of modern optics*, vol. 47, no. 2-3, pp. 415–451, 2000.

[16] D Jaksch, J. Cirac, P Zoller, S. Rolston, R Côté, and M. Lukin, "Fast quantum gates for neutral atoms," *Physical Review Letters*, vol. 85, no. 10, p. 2208, 2000.

[17] E. Knill and R. Laflamme, "Theory of quantum error-correcting codes," *Physical Review A*, vol. 55, no. 2, p. 900, 1997.

[18] D. Gottesman, "Stabilizer codes and quantum error correction," *arXiv preprint quant-ph/9705052*, 1997.

[19] P Aliferis, D Gottesman, and J Preskill, "Quantum accuracy threshold for concatenated distance-3 codes," *Quantum Inf. Comput.*, vol. 6, no. quant-ph/0504218, pp. 97–165, 2005.

[20] R. Gallager, "Low-density parity-check codes," *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.

[21] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge university press, 2008.

[22] M. Li, M. Gutiérrez, S. E. David, A. Hernandez, and K. R. Brown, "Fault tolerance with bare ancillary qubits for a [[7, 1, 3]] code," *Physical Review A*, vol. 96, no. 3, p. 032 341, 2017.

[23] E. Knill, "Quantum computing with realistically noisy devices," *Nature*, vol. 434, no. 7029, pp. 39–44, 2005.

[24] R. Raussendorf and J. Harrington, "Fault-tolerant quantum computation with high threshold in two dimensions," *Physical review letters*, vol. 98, no. 19, p. 190 504, 2007.

[25] A. G. Fowler, A. M. Stephens, and P. Groszkowski, "High-threshold universal quantum computation on the surface code," *Physical Review A*, vol. 80, no. 5, p. 052 312, 2009.

[26] S. Janardan, Y. Tomita, M. Gutiérrez, and K. R. Brown, "Analytical error analysis of clifford gates by the fault-path tracer method," *Quantum Information Processing*, vol. 15, no. 8, pp. 3065–3079, 2016.

[27] S. Bravyi and A. Vargo, "Simulation of rare events in quantum error correction," *Physical Review A*, vol. 88, no. 6, p. 062 308, 2013.

[28] J. Preskill, "Fault tolerant quantum computation," in *Introduction to quantum computation and information*, Singapore: World Scientific, 1998, ch. 8, pp. 213–269.

[29] P. W. Shor, "Fault-tolerant quantum computation," ser. FOCS '96, IEEE Computer Society Press, 1996, pp. 56–65.

[30] P. Aliferis and A. W. Cross, "Subsystem fault tolerance with the bacon-shor code," *Phys. Rev. Lett.*, vol. 98, no. 22, p. 220 502, 2007.

[31] Y. Tomita and K. M. Svore, "Low-distance surface codes under realistic quantum noise," *Physical Review A*, vol. 90, no. 6, p. 062 320, 2014.

[32] A. M. Steane, "Active stabilization, quantum computation, and quantum state synthesis," *Physical Review Letters*, vol. 78, no. 11, p. 2252, 1997.

[33] D. Bacon, "Operator quantum error-correcting subsystems for self-correcting quantum memories," *Physical Review A*, vol. 73, no. 1, p. 012 340, 2006.

[34] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Phys. Rev. A*, vol. 86, no. 3, p. 032 324, 2012.

[35] H. Bombin and M. A. Martin-Delgado, "Topological quantum distillation," *Physical review letters*, vol. 97, no. 18, p. 180 501, 2006.

[36] A. J. Landahl, J. T. Anderson, and P. R. Rice, "Fault tolerant quantum computing with color codes," *arXiv:1108.5738 [quantph]*, 2011.

[37] A. Cross, Personal Communication, 2016.

[38] K. Mølmer and A. Sørensen, "Multiparticle entanglement of hot trapped ions," *Physical Review Letters*, vol. 82, no. 9, p. 1835, 1999.

[39] T. J. Yoder and I. H. Kim, "The surface code with a twist," *Quantum*, vol. 1, p. 2, 2017.

[40] R. Chao and B. W. Reichardt, "Quantum error correction with only two extra qubits," *Physical review letters*, vol. 121, no. 5, p. 050 502, 2018.

[41] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Phys. Rev. A*, vol. 70, no. 5, p. 052 328, 2004.

[42] F. J. MacWilliams and N. J. A. Sloane, *The theory of error correcting codes*. Elsevier, 1977.

[43] K. M. Svore, D. P. Divincenzo, and B. M. Terhal, "Noise threshold for a fault-tolerant two-dimensional lattice architecture," *arXiv preprint quant-ph/0604090*, 2006.

[44] M. Li, D. Miller, and K. R. Brown, "Direct measurement of bacon-shor code stabilizers," *Physical Review A*, vol. 98, no. 5, p. 050 301, 2018.

[45] J. Napp and J. Preskill, "Optimal bacon-shor codes," *Quantum Info. Comput.*, vol. 13, no. 5-6, pp. 490–510, May 2013.

[46] T. J. Yoder, "Universal fault-tolerant quantum computation with bacon-shor codes," *arXiv:1710.01378*, 2017.

[47] H Bombin and M. A. Martin-Delgado, "Optimal resources for topological two-dimensional stabilizer codes: Comparative study," *Physical Review A*, vol. 76, no. 1, p. 012 305, 2007.

[48] Z. Nussinov and J. Van Den Brink, "Compass models: Theory and physical motivations," *Rev. Mod. Phys.*, vol. 87, no. 1, p. 1, 2015.

[49] A. Kitaev, "Anyons in an exactly solved model and beyond," *Annals of Physics*, vol. 321, no. 1, pp. 2–111, 2006.

[50] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4452–4505, 2002.

[51] D. P. DiVincenzo and P. Aliferis, "Effective fault-tolerant quantum computation with slow measurements," *Physical review letters*, vol. 98, no. 2, p. 020 501, 2007.

[52] R. Raussendorf, J. Harrington, and K. Goyal, "Topological fault-tolerance in cluster state quantum computation," *New Journal of Physics*, vol. 9, no. 6, p. 199, 2007.

[53] C. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, "Surface code quantum computing by lattice surgery," *New J. Phys.*, vol. 14, no. 12, p. 123 011, 2012.

[54] R Versluis, S Poletto, N Khammassi, B Tarasinski, N Haider, D. Michalak, A Bruno, K Bertels, and L DiCarlo, "Scalable quantum circuit and control for a superconducting surface code," *Phys. Rev. Appl.*, vol. 8, no. 3, p. 034 021, 2017.

[55] C. J. Trout, M. Li, M. Gutiérrez, Y. Wu, S.-T. Wang, L. Duan, and K. R. Brown, "Simulating the performance of a distance-3 surface code in a linear ion trap," *New Journal of Physics*, vol. 20, no. 4, p. 043 038, 2018.

[56] S. B. Bravyi and A. Y. Kitaev, "Quantum codes on a lattice with boundary," *arXiv:quant-ph/9811052*, 2011.

[57] N. M. Linke, M. Gutierrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe, "Fault-tolerant quantum error detection," *Sci. Adv.*, vol. 3, no. 10, e1701074, 2017.

[58] M. Takita, A. W. Cross, A. Córcoles, J. M. Chow, and J. M. Gambetta, "Experimental demonstration of fault-tolerant state preparation with superconducting qubits," *Physical review letters*, vol. 119, no. 18, p. 180 501, 2017.

[59] A. Sørensen and K. Mølmer, "Quantum computation with ions in thermal motion," *Physical review letters*, vol. 82, no. 9, p. 1971, 1999.

[60] D. Maslov, "Basic circuit compilation techniques for an ion-trap quantum machine," *New J. Phys.*, vol. 19, no. 2, p. 023 035, 2017.

[61] C. Ballance, T. Harty, N. Linke, M. Sepiol, and D. Lucas, "High-fidelity quantum logic gates using trapped-ion hyperfine qubits," *Physical review letters*, vol. 117, no. 6, p. 060 504, 2016.

[62] P. H. Leung, K. A. Landsman, C. Figgatt, N. M. Linke, C. Monroe, and K. R. Brown, "Robust 2-qubit gates in a linear ion crystal using a frequency-modulated driving force," *Physical review letters*, vol. 120, no. 2, p. 020 501, 2018.

[63] M. Li, D. Miller, M. Newman, Y. Wu, and K. R. Brown, "2d compass codes," *Physical Review X*, vol. 9, no. 2, p. 021 041, 2019.

[64] D. M. Debroy, M. Li, S. Huang, and K. R. Brown, "Logical performance of 9 qubit compass codes in ion traps with crosstalk errors," *arXiv preprint arXiv:1910.08495*, 2019.

[65] C. Flühmann, T. L. Nguyen, M. Marinelli, V. Negnevitsky, K. Mehta, and J. Home, "Encoding a qubit in a trapped-ion mechanical oscillator," *Nature*, vol. 566, no. 7745, p. 513, 2019.

[66] D. Riste, S. Poletto, M.-Z. Huang, A. Bruno, V. Vesterinen, O.-P. Saira, and L. DiCarlo, "Detecting bit-flip errors in a logical qubit using stabilizer measurements," *Nature communications*, vol. 6, p. 6983, 2015.

[67] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, J. Heinsoo, J.-C. Besse, M. Gabureac, A. Wallraff, and C. Eichler, "Entanglement stabilization using ancilla-based parity detection and real-time feedback in superconducting circuits," *npj Quantum Information*, vol. 5, no. 1, pp. 1–7, 2019.

[68] V. Negnevitsky, M. Marinelli, K. K. Mehta, H.-Y. Lo, C. Flühmann, and J. P. Home, "Repeated multi-qubit readout and feedback with a mixed-species trapped-ion register," *Nature*, vol. 563, no. 7732, p. 527, 2018.

[69] D. G. Cory, M. Price, W Maas, E Knill, R. Laflamme, W. H. Zurek, T. F. Havel, and S. Somaroo, "Experimental quantum error correction," *Physical Review Letters*, vol. 81, no. 10, p. 2152, 1998.

[70] J Chiaverini, D Leibfried, T. Schaetz, M. D. Barrett, R. Blakestad, J Britton, W. M. Itano, J. D. Jost, E Knill, C Langer, *et al.*, "Realization of quantum error correction," *Nature*, vol. 432, no. 7017, p. 602, 2004.

[71] P. Schindler, J. T. Barreiro, T. Monz, V. Nebendahl, D. Nigg, M. Chwalla, M. Hennrich, and R. Blatt, "Experimental repetitive quantum error correction," *Science*, vol. 332, no. 6033, pp. 1059–1061, 2011.

[72] T. H. Taminiau, J. Cramer, T. van der Sar, V. V. Dobrovitski, and R. Hanson, "Universal control and error correction in multi-qubit spin registers in diamond," *Nat. Nanotechnol.*, vol. 9, no. 3, p. 171, 2014.

[73] S. Rosenblum, P Reinhold, M. Mirrahimi, L. Jiang, L Frunzio, and R. Schoelkopf, "Fault-tolerant detection of a quantum error," *Science*, vol. 361, no. 6399, pp. 266–270, 2018.

[74] J. Kelly, R Barends, A. Fowler, A Megrant, E Jeffrey, T. White, D Sank, J. Mutus, B Campbell, Y. Chen, *et al.*, "State preservation by repetitive error detection in a superconducting quantum circuit," *Nature*, vol. 519, no. 7541, p. 66, 2015.

[75] R. Harper and S. T. Flammia, "Fault-tolerant logical gates in the ibm quantum experience," *Physical review letters*, vol. 122, no. 8, p. 080 504, 2019.

[76] K Wright, K. Beck, S Debnath, J. Amini, Y Nam, N Grzesiak, J.-S. Chen, N. Pisenti, M Chmielewski, C Collins, *et al.*, "Benchmarking an 11-qubit quantum computer," *arXiv preprint arXiv:1903.08181*, 2019.

[77] T. E. O'Brien, B. Tarasinski, and L. DiCarlo, "Density-matrix simulation of small surface codes under current and projected experimental noise," *npj Quantum Inf.*, vol. 3, no. 1, p. 39, 2017.

[78] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical review A*, vol. 52, no. 4, R2493, 1995.

[79] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Physical Review A*, vol. 54, no. 2, p. 1098, 1996.

[80] A. M. Steane, "Error correcting codes in quantum theory," *Physical Review Letters*, vol. 77, no. 5, p. 793, 1996.

[81] D. Litinski, "A game of surface codes: Large-scale quantum computing with lattice surgery," *arXiv preprint arXiv:1808.02892*, 2018.

[82] P. Iyer and D. Poulin, "A small quantum computer is needed to optimize fault-tolerant protocols," *Quantum Science and Technology*, vol. 3, no. 3, p. 030 504, 2018.

[83] R. Harper, S. T. Flammia, and J. J. Wallman, "Efficient learning of quantum noise," *arXiv preprint arXiv:1907.13022*, 2019.

[84] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Physical Review A*, vol. 54, no. 5, p. 3824, 1996.

[85] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, "Perfect quantum error correcting code," *Physical Review Letters*, vol. 77, no. 1, p. 198, 1996.

[86] A. M. Steane, "Error correcting codes in quantum theory," *Physical Review Letters*, vol. 77, no. 5, p. 793, 1996.

[87] A. Robertson, C. Granade, S. D. Bartlett, and S. T. Flammia, "Tailored codes for small quantum memories," *Physical Review Applied*, vol. 8, no. 6, p. 064 004, 2017.

[88] B. W. Reichardt, "Fault-tolerant quantum error correction for steane's seven-qubit color code with few or no extra qubits," *arXiv preprint arXiv:1804.06995*, 2018.

[89] C. Chamberland and M. E. Beverland, "Flag fault-tolerant error correction with arbitrary distance codes," *Quantum*, vol. 2, no. 53, pp. 10–22 331, 2018.

[90] L. Lao and C. G. Almudever, "Fault-tolerant quantum error correction on near-term quantum processors using flag and bridge qubits," *arXiv preprint arXiv:1909.07628*, 2019.

[91] S. Olmschenk, K. C. Younge, D. L. Moehring, D. N. Matsukevich, P. Maunz, and C. Monroe, "Manipulation and detection of a trapped yb^+ hyperfine qubit," *Physical Review A*, vol. 76, no. 5, p. 052 314, 2007.

[92] R. Noek, G. Vrijsen, D. Gaultney, E. Mount, T. Kim, P. Maunz, and J. Kim, "High speed, high fidelity detection of an atomic hyperfine qubit," *Optics letters*, vol. 38, no. 22, pp. 4735–4738, 2013.

[93] E. Mount, C. Kabytayev, S. Crain, R. Harper, S.-Y. Baek, G. Vrijsen, S. T. Flammia, K. R. Brown, P. Maunz, and J. Kim, "Error compensation of single-qubit gates in a surface-electrode ion trap using composite pulses," *Physical Review A*, vol. 92, no. 6, p. 060 301, 2015.

[94] Y. Wu, S.-T. Wang, and L.-M. Duan, "Noise analysis for high-fidelity quantum entangling gates in an anharmonic linear paul trap," *Physical Review A*, vol. 97, no. 6, p. 062 325, 2018.

[95] J. T. Merrill, S. C. Doret, G. Vittorini, J. Addison, and K. R. Brown, "Transformed composite sequences for improved qubit addressing," *Physical Review A*, vol. 90, no. 4, p. 040 301, 2014.

[96] J. Chiaverini, D. Leibfried, T. Schaetz, M. D. Barrett, R. B. Blakestad, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer, R. Ozeri, and D. J. Wineland, "Realization of quantum error correction," *Nature*, vol. 432, no. 7017, pp. 602–605, Dec. 2004.

[97] D. Nigg, M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt, "Quantum computations on a topologically encoded qubit," *Science*, vol. 345, no. 6194, pp. 302–305, 2014.

[98] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, and D. M. Meekhof, "Experimental issues in coherent quantum-state manipulation of trapped atomic ions," *J. Res. of the Natl. Inst. of Stand. Tech.*, vol. 103, no. 3, pp. 259–328, 1998.

[99] D. Kielpinski, C. Monroe, and D. J. Wineland, "Architecture for a large-scale ion-trap quantum computer," *Nature*, vol. 417, no. 6890, pp. 709–711, 2002.

[100] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, and W. K. Hensinger, *Sci. Adv.*, vol. 3, no. 2, 2017.

[101] D. L. Moehring, P. Maunz, S. Olmschenk, K. C. Younge, D. N. Matsukevich, L.-M. Duan, and C. Monroe, "Entanglement of single-atom quantum bits at a distance," *Nature*, vol. 449, no. 7158, pp. 68–71, 2007.

[102] L.-M. Duan and C. Monroe, "Colloquium: Quantum networks with trapped ions," *Reviews of Modern Physics*, vol. 82, no. 2, p. 1209, 2010.

[103] N. C. Brown and K. R. Brown, "Comparing zeeman qubits to hyperfine qubits in the context of the surface code: Yb+ 174 and yb+ 171," *Physical Review A*, vol. 97, no. 5, p. 052 301, 2018.

[104] M Gutiérrez, M Müller, and A Bermudez, "Transversality and lattice surgery: Exploring realistic routes toward coupled logical qubits with trapped-ion quantum processors," *Physical Review A*, vol. 99, no. 2, p. 022 330, 2019.

[105] A Bermudez, X Xu, M Gutiérrez, S. Benjamin, and M Müller, "Fault-tolerant protection of near-term trapped-ion topological qubits under realistic noise sources," *arXiv preprint arXiv:1810.09199*, 2018.

[106] P. H. Leung and K. R. Brown, "Entangling an arbitrary pair of qubits in a long ion crystal," *Physical Review A*, vol. 98, no. 3, p. 032 318, 2018.

[107] S.-L. Zhu, C. Monroe, and L.-M. Duan, "Arbitrary-speed quantum gates within large ion crystals through minimum control of laser beams," *EPL (Europhysics Letters)*, vol. 73, no. 4, p. 485, 2006.

[108] R. Blumel, N. Grzesiak, and Y. Nam, "Power-optimal, stabilized entangling gate between trapped-ion qubits," *arXiv preprint arXiv:1905.09292*, 2019.

[109] K. A. Landsman, Y. Wu, P. H. Leung, D. Zhu, N. M. Linke, K. R. Brown, L. Duan, and C. R. Monroe, "Two-qubit entangling gates within arbitrarily long chains of trapped ions," *arXiv preprint arXiv:1905.10421*, 2019.

[110] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov, and C. Monroe, "Parallel entangling operations on a universal ion-trap quantum computer," *Nature*, vol. 572, no. 7769, pp. 368–372, 2019.

[111] K. R. Brown, A. W. Harrow, and I. L. Chuang, "Arbitrarily accurate composite pulse sequences," *Physical Review A*, vol. 70, no. 5, p. 052 318, 2004.

[112] D. M. Debroy, M. Li, M. Newman, and K. R. Brown, "Stabilizer slicing: Coherent error cancellations in low-density parity-check stabilizer codes," *Physical review letters*, vol. 121, no. 25, p. 250 502, 2018.

[113] Z. Cai, X. Xu, and S. C. Benjamin, "Mitigating coherent noise using pauli conjugation," *arXiv preprint arXiv:1906.06270*, 2019.

[114] L. Viola, E. Knill, and S. Lloyd, "Dynamical decoupling of open quantum systems," *Physical Review Letters*, vol. 82, no. 12, p. 2417, 1999.

[115] R. V. Book *et al.*, "Michael r. garey and david s. johnson, computers and intractability: A guide to the theory of np-completeness," *Bulletin (New Series) of the American Mathematical Society*, vol. 3, no. 2, pp. 898–904, 1980.

[116] R. E. Bellman, "Dynamic programming treatment of the traveling salesman problem.," 1961.

[117] D Kielpinski, B. King, C. Myatt, C. Sackett, Q. Turchette, W. M. Itano, C Monroe, D. J. Wineland, and W. Zurek, "Sympathetic cooling of trapped ions for quantum logic," *Physical Review A*, vol. 61, no. 3, p. 032 310, 2000.

[118] M. D. Barrett, B DeMarco, T Schaetz, V Meyer, D Leibfried, J Britton, J Chiaverini, W. Itano, B Jelenković, J. Jost, *et al.*, "Sympathetic cooling of $^{9}\mathrm{Be}^{+}$ and $^{2}4\mathrm{Mg}^{+}$ for quantum logic," *Physical Review A*, vol. 68, no. 4, p. 042 302, 2003.

[119] J. P. Home, M. McDonnell, D. Szwer, B. Keitch, D. Lucas, D. Stacey, and A. Steane, "Memory coherence of a sympathetically cooled trapped-ion qubit," *Physical Review A*, vol. 79, no. 5, p. 050 305, 2009.

[120] Y. Wang, M. Um, J. Zhang, S. An, M. Lyu, J.-N. Zhang, L.-M. Duan, D. Yum, and K. Kim, "Single-qubit quantum memory exceeding ten-minute coherence time," *Nature Photonics*, vol. 11, no. 10, p. 646, 2017.

[121] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, "Topological and subsystem codes on low-degree graphs with flag qubits," *arXiv preprint arXiv:1907.09528*, 2019.

[122] D. S. Wang, A. G. Fowler, and L. C. Hollenberg, "Surface code quantum computing with error rates over 1%," *Physical Review A*, vol. 83, no. 2, p. 020 302, 2011.

[123] S. Bravyi, D. Poulin, and B. Terhal, "Tradeoffs for reliable quantum information storage in 2d systems," *Physical review letters*, vol. 104, no. 5, p. 050 503, 2010.

[124] D. Gottesman, "Fault-tolerant quantum computation with constant overhead," *Quantum Information & Computation*, vol. 14, no. 15-16, pp. 1338–1372, 2014.

[125] J.-P. Tillich and G. Zémor, "Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1193–1202, 2013.

[126] A. Leverrier, J.-P. Tillich, and G. Zémor, "Quantum expander codes," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, IEEE, 2015, pp. 810–824.

[127] O. Fawzi, A. Grospellier, and A. Leverrier, "Constant overhead quantum fault-tolerance with quantum expander codes," in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE, 2018, pp. 743–754.

[128] A. Grospellier and A. Krishna, "Numerical study of hypergraph product codes," *arXiv preprint arXiv:1810.03681*, 2018.

[129] M. Sipser and D. A. Spielman, "Expander codes," *IEEE transactions on Information Theory*, vol. 42, no. 6, pp. 1710–1722, 1996.

[130] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 599–618, 2001.

[131] S. Bravyi, G. Duclos-Cianci, D. Poulin, and M. Suchara, "Subsystem surface codes with three-qubit check operators," *arXiv preprint arXiv:1207.1443*, 2012.

[132] T. J. Yoder, "Optimal quantum subsystem codes in two dimensions," *Physical Review A*, vol. 99, no. 5, p. 052 333, 2019.

[133] D. Bacon and A. Casaccino, "Quantum error correcting subsystem codes from two classical linear codes," *arXiv preprint quant-ph/0610088*, 2006.

[134] D. A. Spielman, "Linear-time encodable and decodable error-correcting codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1723–1731, 1996.

[135] O. Fawzi, A. Grospellier, and A. Leverrier, "Efficient decoding of random errors for quantum expander codes," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, ACM, 2018, pp. 521–534.

[136] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of provably good low-density parity check codes," in *2000 IEEE International Symposium on Information Theory (Cat. No. 00CH37060)*, IEEE, 2000, p. 199.

[137] V. Guruswami, "Iterative decoding of low-density parity check codes (a survey)," *arXiv preprint cs/0610022*, 2006.

[138] D. Burshtein and G. Miller, "Expander graph arguments for message-passing algorithms," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 782–790, 2001.

[139] D. J. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[140] N. Delfosse and N. H. Nickerson, "Almost-linear time decoding algorithm for topological codes," *arXiv preprint arXiv:1709.06218*, 2017.

[141] P. Aliferis and J. Preskill, "Fault-tolerant quantum computation against biased noise," *Physical Review A*, vol. 78, no. 5, p. 052 331, 2008.

[142] D. K. Tuckett, S. D. Bartlett, and S. T. Flammia, "Ultrahigh error threshold for surface codes with biased noise," *Physical review letters*, vol. 120, no. 5, p. 050 505, 2018.

[143] A. M. Stephens, W. J. Munro, and K. Nemoto, "High-threshold topological quantum error correction against biased noise," *Physical Review A*, vol. 88, no. 6, p. 060 301, 2013.

[144] N. Delfosse and J.-P. Tillich, "A decoding algorithm for css codes using the x/z correlations," in *2014 IEEE International Symposium on Information Theory*, IEEE, 2014, pp. 1071–1075.

[145] N. H. Nickerson and B. J. Brown, "Analysing correlated noise on the surface code using adaptive decoding algorithms," *Quantum*, vol. 3, p. 131, 2019.

[146] A. S. Darmawan and D. Poulin, "Linear-time general decoding algorithm for the surface code," *Physical Review E*, vol. 97, no. 5, p. 051 302, 2018.

[147] N. Maskara, A. Kubica, and T. Jochym-O'Connor, "Advantages of versatile neural-network decoding for topological codes," *Physical Review A*, vol. 99, no. 5, p. 052 351, 2019.